

BAYESIAN MACHINE LEARNING

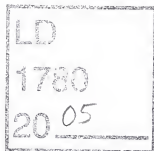
By

SOUNAK CHAKRABORTY

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2005



, C435

Copyright 2005
by
Sounak Chakraborty

To my mother and my father

ACKNOWLEDGMENTS

First, I would like to thank my mother and my father for all their love and support shown at all stages of my life.

I express my warmest gratitude to my academic adviser and committee chair, Dr. Malay Ghosh. He has been a great companion during the ups and downs of my journey toward the Ph. D. : Learning from his intuition and his clear understanding of mathematical statistics has always been a pleasure.

I would also like to extend thanks to my other supervisory committee members: Dr. Andrew Rosalsky, Dr. James P. Hobert, Dr. Michael Daniels, and Dr. Anurag Agarwal. I also thank all of the faculty of the Department of Statistics, University of Florida; whenever I needed advice or feedback, I felt welcome in their offices. In addition, thanks go to Dr. Bani K. Mallick of Texas A & M University, Dr. Tapabrata Maiti of Iowa State University, and Dr. Debashish Ghosh of University of Michigan for all the datasets they provided and their helpful suggestions during all stages of my research.

Much of my training in statistics goes back to St. Xavier's College, Calcutta where I earned my bachelor's degree; and the Indian Statistical Institute, Calcutta where I pursued my master's degree. I would like to thank all of my professors back in India who actually sowed the first seeds of interest in statistics in me. Two individuals deserve special mention : Dr. A.M. Goon and Dr. J.K. Ghosh have been my inspiration in this journey.

Equally important have been my friends at the University of Florida with their humor, comradeship, and unwavering support. Thanks go to Dola Kakima for showing all the love and care, and making me feel at home. My heartfelt thanks

go to Dr. Bhramar Mukherjee for her unstinting counsel and encouragement during some of my toughest times. I will always recall with fondness the good times spent here during my graduate work.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	x
ABSTRACT	xi
CHAPTER	
1 INTRODUCTION	1
1.1 Bayesian vs Frequentist Method of Learning	3
1.2 Curse of Dimensionality	5
1.3 Feedforward Neural Networks	7
1.4 Support Vector Machine	13
1.5 Multiple Linear Regression	21
1.6 Principal Component Regression	22
1.7 Partial Least Squares	23
1.8 Random Forest	25
1.9 Motivation and Outline of the Study	26
2 BAYESIAN NEURAL NETWORK FOR BIVARIATE BINARY DATA AND ITS APPLICATION TO PROSTATE CANCER STUDY	29
2.1 Prostate Cancer	30
2.2 Standard Methods	31
2.2.1 Bivariate Logit Model	31
2.2.2 Neural Networks	32
2.3 Hierarchical Bayesian Neural Network Model for Bivariate Binary Data	32
2.4 Applications to Prostate Cancer Study	37
2.4.1 Example 1: Application for Prostate Cancer with Clinical Covariates	37
2.4.2 Example 2: Application for Gene Expression Microarray Data	43
2.4.3 Simulation Study	47
2.5 Summary and Conclusion	48

3	BAYESIAN NONLINEAR REGRESSION FOR LARGE p SMALL n PROBLEMS	51
3.1	Regression Method Based on Reproducing Kernel Hilbert Space	55
3.2	Hierarchical Bayes Relevance Vector Machine	56
3.3	Hierarchical Bayes Support Vector Machine	60
3.4	Hierarchical Bayes RVM for Multivariate Regression	65
3.5	Hierarchical Bayes SVM for Multivariate Regression	67
3.6	Prediction	70
3.7	Case Study	71
3.7.1	Application to Fluorescence Spectra	71
3.7.2	Application to NIR Spectroscopy Of Biscuit Doughs	75
3.7.3	Simulation Study	80
3.8	Empirical Bayes Support Vector Machine And Relevance Vector Machine	82
3.9	Concluding Remarks	84
4	MULTICLASS CANCER DIAGNOSIS USING BAYESIAN KERNEL MACHINE MODELS	89
4.1	Gene Expression Microarrays	92
4.2	RKHS based Bayesian Multinomial logit model	93
4.3	Bayesian Multicategory Support Vector Machine	99
4.4	A Bayesian Gene Selection Scheme	103
4.5	Classification of Future Cases and Identifying the Significant Genes	106
4.6	Applications in Multiclass Cancer Classification Using Microarrays	107
4.6.1	Golub Leukemia Data	109
4.6.2	Glioma Cancer Data	111
4.7	Discussion	115
5	SUMMARY, DISCUSSION AND FUTURE RESEARCH	119
5.1	Future Research	122
5.1.1	Neural Networks with Variable Number of Hidden Nodes	122
5.1.2	Bayesian Gene Selection Model for Neural Networks	122
5.1.3	Nonparametric and Semiparametric Bayes Models for Tumor Classification	122
5.1.4	Applications in Proteomics and Spatio-temporal Data	123
	REFERENCES	124
	BIOGRAPHICAL SKETCH	130

LIST OF TABLES

<u>Table</u>		<u>page</u>
2-1	Percentage of correct prediction in the 234 test cases and 300 validation cases using our bivariate Hierarchical Bayesian Neural Network (HBNN) in Example 1.	41
2-2	Percentage of correct prediction in the 234 test cases and 300 validation cases using our bivariate Hierarchical Bayesian Neural Network (HBNN) as well as Ripley's Classical Neural Network (Ripley), Neal's Bayesian Neural Network (Neal), Bivariate Logistic Regression Models (BLM), and Univariate Hierarchical Bayesian Neural Network (UHBNN) in Example 1.	41
2-3	Leave one out cross-validation error of prediction using our bivariate Hierarchical Bayesian Neural Network (HBNN) in Example 2.	46
2-4	Leave one out cross-validation error using our bivariate Hierarchical Bayesian Neural Network (HBNN) as well as Ripley's Classical Neural Network (Ripley), Neal's Bayesian Neural Network (Neal), Bivariate Logistic Regression Models (BLM), and Univariate Hierarchical Bayesian Neural Network (UHBNN).	46
2-5	Leave one out cross-validation error of prediction using our bivariate Hierarchical Bayesian Neural Network (HBNN) with clinical covariates in Example 2.	47
2-6	Leave one out cross-validation error of prediction using our bivariate HBNN in Example 2 with 10 hidden nodes.	47
2-7	Average percentage of correct prediction in the 234 test cases and 300 validation cases using our bivariate Hierarchical Bayesian Neural Network (HBNN) in Example 3.	48
2-8	Percentage of correct prediction in the 234 test cases and 300 validation cases using our bivariate Hierarchical Bayesian Neural Network (HBNN) as well as Ripley's Classical Neural Network (Ripley), Neal's Bayesian Neural Network (Neal), Bivariate Logistic Regression Models (BLM), and Univariate Hierarchical Bayesian Neural Network (UHBNN) in Example 3.	49

3-1	Average mean squared errors of prediction (MSEP) on the 9 blood sample in the test set. Results from models with single smoothing parameters are marked by a *.	75
3-2	Mean squared errors of prediction on the 39 biscuit dough pieces in the validation set.	78
3-3	Mean squared errors of prediction on the 39 biscuit dough pieces in the validation set when the sample 23 is included in the training set. . . .	79
3-4	Mean squared errors of prediction in the simulated dataset.	81
4-1	Missclassification error in the Test Set of the Leukemia data.	110
4-2	Total number of missclassifications in the leave one out crossvalidation in the glioma data.	114

LIST OF FIGURES

Figure	page
1-1 Schematic diagram of a single hidden layer neural network.	8
1-2 Separable and nonseparable cases for classification.	14
1-3 Support vector classifier.	15
1-4 (a) Vapnik's ϵ -insensitive ($\epsilon = 1$). (b) Hinge loss.	18
2-1 Trace plots of MCMC samples generated from the posterior distributions of β_1 in one chain for Example 1 and hyperparameter choice (i). . . .	37
2-2 Trace plots of MCMC samples generated from the posterior distributions of γ_1 in one chain for Example 1 and hyperparameter choice (i). . . .	38
2-3 Trace plots of MCMC samples generated from the posterior distributions of γ_2 in one chain for Example 1 and hyperparameter choice (i). . . .	40
2-4 Marginal relevance of genes according to the between and within sum of squares criteria (BWSS)	44
3-1 The figure in the left hand side represents Vapnik's ϵ -insensitive ($\epsilon = 1$) loss function. The figure in the right hand side represents the corresponding likelihood.	61
3-2 Our ϵ -insensitive loss function when $q = 2$ and $\rho_1 = \rho_2 = 1$ ($\epsilon = 1$). . .	68
3-3 Optical spectra for 101 wavelengths on the bloodsugar data.	72
3-4 Average MSEP in the test set for different choices of ϵ in the bloodsugar data.	73
3-5 Blood sugar concentration data set.	86
3-6 Biscuit dough composition data set.	87
4-1 A schematic diagram of production of protein from DNA.	93
4-2 Heatmap of Golub leukemia DNA data.	94
4-3 Selected genes in Golub leukemia DNA data.	112
4-4 Selected genes in glioma DNA data.	116

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

BAYESIAN MACHINE LEARNING

By

Sounak Chakraborty

August 2005

Chair: Malay Ghosh
Major Department: Statistics

Statistical machine learning has seen a lot of progress in the last two decades. Rather than being a purely algorithmic approach to machine learning, the statistical methodology of Bayesian learning is distinguished by its use of probability to express all forms of uncertainty. Learning and other forms of inference can then be performed by simply applying the rules of probability. Results of Bayesian learning are expressed in terms of a probability distribution over all unknown quantities. These probabilities can be interpreted only as expressions of our degree of belief in the various possibilities. In contrast, the conventional frequentist strategy takes the form of an estimator for unknown quantities, with possibly some optimality criterion in mind.

We extended several machine learning methods (neural network, support vector machine, and relevance vector machine) to the Bayesian paradigm. We developed a hierarchical Bayesian neural network for binary and bivariate binary data. The main applications of our developed methodology are in staging and classification of prostate cancer. Several applications using clinical and gene expression microarray covariates along with simulations show empirically that our proposed method is more

accurate in predicting stages of prostate cancer compared to other methods (Neal's Bayesian neural network, logistic regression, and the classical neural network). We also proposed a Bayesian support vector machine (SVM) and a relevance vector machine (RVM) to tackle the problems of nonlinear regression and classification in "high dimensional low sample size" data. Our Bayesian SVM and RVM constructed on reproducing kernel Hilbert space theory does not require an additional projection into the sample space, and gives a stronger learning algorithm than the classical one. Its virtue over the standard methods is that it produced a richer class of models that provide better predictions and it has the unique ability to quantify prediction errors. Several applications on near-infrared spectroscopy data and gene expression microarray data indicate that our Bayesian SVM and RVM performs better than most of the available models.

CHAPTER 1 INTRODUCTION

“One machine can do the work of fifty ordinary men. No machine can do the work of one extraordinary man.” —Elbert Hubbard

Humans are capable of tackling extremely difficult problems without the benefit of an a priori solution. We learn from experience, and can often transfer knowledge acquired to novel instances or even whole new tasks. Are machines capable of similar problem-solving prowess? Machine learning is a direct descendant of an older discipline — statistical model fitting. The goal in machine learning is to extract useful information from a corpus of data, by building good probabilistic models. The particular twist behind machine learning, however is to automate this process as much as possible, often by using very flexible models characterized by large numbers of parameters, and let the machine take care of the rest. Clearly, machine learning is driven by rapid technological progress in two areas: storage devices leading to large databases and data sets, and computing power enabling the use of more complex models.

The Bayesian machine learning uses probability to express all forms of uncertainty. Learning is performed by simple applications of the rules of probability. Results of Bayesian machine learning are expressed in terms of a probability distribution over all unknown quantities or parameters of the model. The probability distribution of a parameter quantifies the uncertainty of its value, and expresses our degree of belief in the various possibilities. In contrast, the conventional frequentist strategy takes the form of an estimator for unknown quantities or model parameters, with possibly some optimality criterion in mind. Section 1.1 briefly illustrates Bayesian and frequentist views of learning.

The term neural network includes a large class of models and learning methods. It was originally developed with the goal of modeling information processing and learning in the brain. At the most basic level, neural networks can be viewed as a broad class of parameterized graphical models consisting of networks with interconnected units. Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A great deal of research has examined neural networks in the last two decades. Although a much hype, glamor, and enthusiasm are, created around its wide-ranging success, yet neural networks are basically analogous to nonlinear statistical models which can be applied to both classification and regression purposes. The neural networks most commonly used are the multilayer perceptron networks, also known as “back-propagation” or “feed-forward” networks (Figure 1-1). In the Bayesian approach to neural network, the objective is to find the predictive distribution for the target values in a new “test” case, given the inputs for that case, and the inputs and targets in the training cases. Section 1.3 gives an introduction to both frequentist and Bayesian neural networks.

Support vector machines (SVM) are machine learning approaches, originally developed by Vapnik (1995) and others who worked in this area. The SVMs are a system for efficiently training the linear learning machines in the kernel induced feature space. It has gained popularity due to its attractive, analytic and computational features, and promising empirical performance. The formulation embodies the Structural Risk Minimization (SRM) principle which has been shown to be superior to the Empirical Risk Minimization (ERM), used by conventional neural networks. It has been also shown that SVM methodology can be cast as a regularization problem in terms of reproducing kernel Hilbert space (RKHS) (Hastie et al., 2001). Hence SVM and penalty methods (as used in statistical theory for

nonparametric regression) have a strong relationship. Sollich (1999) showed that SVM can be interpreted as a maximum a posterior solution to inference problems with Gaussian priors and an appropriate likelihood function based on a probabilistic interpretation. Section 1.4 explains SVMs in frequentist and Bayesian paradigms.

Several other methods like multiple linear regression, principal component regression, partial least squares, and random forest are briefly described in Sections 1.5 to 1.8.

1.1 Bayesian vs Frequentist Method of Learning

In learning, we combine our prior knowledge of the problem with the information in the training data. One major trend in learning research is to reason with probabilities, in order to model the uncertainty present in all of our models and data. Probabilistic modeling provides a very powerful, general-purpose tool for expressing relative certainty in our understanding of the world. Probability theory provides a principled mechanism for reasoning with uncertainty, learning from data, and generating new data by sampling from a learned model.

Let $x_1, x_2, \dots$ be a set of independent observations generated by some underlying process. We can also assign a probability model to these x_i with some unknown model parameter θ . Let us denote such probability distribution or densities by $P(x_i|\theta)$. If our assumption about the underlying probability distribution is correct, then learning about the whole underlying process will be learning about the underlying parameter θ , which completely characterizes the underlying probability distribution. Say, we have observed some of the independent observations x_i , as $x_1, x_2, \dots, x_n$. The joint probability distribution of $x_1, x_2, \dots, x_n$ is also known as the likelihood function, is shown in Equation 1-1.

$$\begin{aligned} L(\theta) &= L(\theta|x_1, x_2, \dots, x_n) \\ &\propto P(x_1, x_2, \dots, x_n|\theta) = \prod_{i=1}^n P(x_i|\theta) \end{aligned} \tag{1-1}$$

Loosely speaking, the likelihood of a set of data is the probability of obtaining that particular set of data, given the chosen probability distribution model. In other words we can also say a likelihood function shows the impact of the observations in our hand under a particular probability model assumption.

Maximum likelihood estimation aims to find the most “likely” values of distribution parameter θ for a set of data $x_1, x_2, \dots, x_n$ by maximizing the likelihood function. The estimate of the unknown parameter θ obtained by maximizing the likelihood function $L(\theta)$ is denoted by $\hat{\theta}$.

Most problems in machine learning concerns the value of some quantity that may be observed in the future, say x_{n+1} . In a frequentist setup to predict the future observation x_{n+1} , we use the estimated value of θ , i.e., $\hat{\theta}$, and hence use $P(x_{n+1}|\hat{\theta})$ to make a prediction.

In Bayesian thinking we consider not only what the data has to say (i.e the likelihood), but also what our expertise tells us. So in Bayesian form of learning, we assign a probability distribution to the unknown model parameters which is known as the prior distributions $P(\theta)$. This $P(\theta)$ is what we know about θ before the data are collected and expresses our belief of how likely the different parameter values are. After we observe the data $x_1, x_2, \dots, x_n$, we update this prior distribution to posterior distribution by the Bayes rule. The Bayes rule is calculated by multiplying the prior probability distribution by the likelihood function, and then dividing by the normalizing constant, as in Equation 1-2,

$$P(\theta|x_1, x_2, \dots, x_n) = \frac{P(x_1, x_2, \dots, x_n|\theta)P(\theta)}{\int P(x_1, x_2, \dots, x_n|\theta)P(\theta)d\theta} \quad (1-2)$$

$$\propto L(\theta|x_1, x_2, \dots, x_n)P(\theta)$$

Thus the posterior probability distribution given in Equation 1-2 is the conditional probability distribution of the model parameter given the data. In Bayesian paradigm all further inference about θ are based on this posterior distribution. So we see that

by introducing priors we are able to move from the likelihood function in Equation 1-1, which was not a probability distribution to the posterior distribution which is a probability distribution given in Equation 1-2.

Prediction of the future observation x_{n+1} is based on the posterior predictive distribution (out-of-sample) as shown in Equation 1-3.

$$P(x^{new}|x_1, x_2, \dots, x_n) = \int P(x^{new}|\theta)P(\theta|x_1, x_2, \dots, x_n)d\theta \quad (1-3)$$

This posterior predictive distribution gives a much more detailed view of our predicted x_{n+1} , because now we can also quantify the error by measuring how much mass is given around a particular predicted value. For this reason it is also known as the “complete Bayesian inference” regarding x^{n+1} . In contrast, in a frequentist setup, we use a kind of plug-in distribution rather than integrating out the parameter. This is where the Bayesian approach scores over the frequentist method.

1.2 Curse of Dimensionality

Bellman (1961) introduced the term “Curse of Dimensionality” (COD) in the context of approximation theory. The term implies that prediction and fitting becomes very tough as the dimension of the space increases. Specifically, the COD implies that the mean integrated squared error increases faster than linearly in p , the dimension.

The COD occurs for three main reasons. First, in high dimensions, datasets are sparse. Suppose that n points are uniformly distributed in a unit hypercube in p -dimensional space. Then the proportion of points one expects in a subcube of side s is s^p . But s^p goes to zero as p increases for s between 0 and 1. So there is no data available for local fitting. Second, in high dimensions, the complexity of the set of possible models increases rapidly. For example, let us just consider quadratic models in p variables. Then the number of possible models is $2^{1+p+\frac{p(p-1)}{2}} - 1$. Hence it is understandable, that a huge amount of data will be required to pick the correct

model. Third, in high dimensions, datasets show multicollinearity. This occurs when the explanatory variables lie nearly upon an affine subspace. When p is large then there are many possible subspaces. Thus just by chance, nearly all the data will lie along one of them. The COD which is in statistics for the last five decades now have some very popular names like, “large p small n problems”, “high dimension and low sample size problems or HDLSS”, and also amusingly some people like to call it “short fat data problem”.

Current technological trends inexorably lead to data flood. More data are generated from banking, telecom, and other business transactions. More data are generated from scientific experiments in astronomy, space exploration, biology, high-energy physics, etc. More data are created on the web, especially in text, image, and other multimedia format. For example, Europe's Very Long Baseline Interferometry (VLBI) has 16 telescopes, each of which produces 1 Gigabit/second of astronomical data over a 25-day observation session. This truly generates an “astronomical” amount of data. AT&T handles so many calls per day that it cannot store all of the data and data analysis must be done “on the fly”. DNA microarray is a revolutionary new technology that allows measurement of gene expression levels for many thousands of genes simultaneously (more about microarrays will be discussed in Chapter 4). Microarrays recently became a popular source of high dimension low sample size data. Some of the typical problems are, given microarray data for a number of patients (samples), can we accurately diagnose the disease, predict outcome for a given treatment, and recommend the best treatment? For example the leukemia data set of Golub has 72 samples, and about 7,000 genes. The samples belong to two classes: Acute Lymphoblastic (ALL) and Acute Myeloid (AML), which look similar under a microscope but have very different genetic expression levels. The best diagnostic model was learned on the training set of 38 samples and applied to

a test set of the remaining 34 samples. In the next sections, we will discuss some methods which can be used to analyze these high dimensional data.

1.3 Feedforward Neural Networks

Neural networks were originally intended as abstract models of the brain. However, over the years, their scope extended enormously, and they are now used in a variety of applications involving regression and classification. A neural network is a set of simple computational units that are highly interconnected (Figure 1-1). These units are also called nodes, and loosely represent the biological neuron. For simplicity here, we consider only “backpropagation” or “feedforward” networks. These networks take on a set of inputs $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$, and compute from them some outputs O_i , using a certain number of layers. In a typical neural network with one hidden layer, the outputs might be computed as shown in Equation 1-4.

$$O_i = b_0 + \sum_{j=1}^M b_j \psi(c_{j0} + \sum_{s=1}^p x_{is} c_{js}), \quad i = 1, \dots, n. \quad (1-4)$$

In Equation 1-4, c_{js} is the weight from input x_{is} to the hidden unit j . Similarly, b_j is the weight attached to the hidden unit j . The c_{j0} and b_0 are the biases for the hidden nodes and the output units. The number of hidden nodes is denoted by M . However, if necessary, they can be absorbed in the c_{js} and the b_j . (Rios Insua and Müller, 1998). The function ψ is referred to as an *activation function*. Typically, ψ is nonlinear. Some of the most common choices of ψ are the logistic and the hyperbolic tangent functions.

Cybenko (1989) showed that if the number of hidden nodes tends to infinity, then these can be used as universal approximator of any continuous function in a compact range. Let I_n denote the n -dimensional unit cube, $[0, 1]^n$. The space of continuous functions on I_n is denoted by $C(I_n)$ and use $\|f\|$ to denote the supremum norm of an $f \in C(I_n)$. Cybenko (1989) showed that when ψ is any continuous sigmoidal function, then sums of the form Equation 1-4 are dense in $C(I_n)$. In a classical approach, the

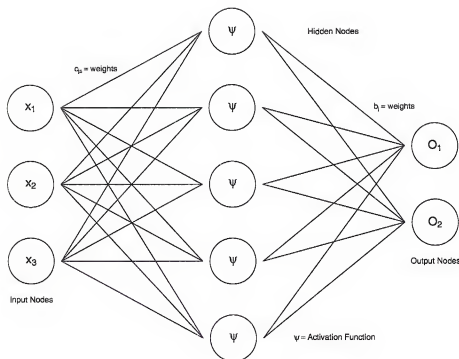


Figure 1-1: Schematic diagram of a single hidden layer neural network.

weights and biases in neural networks are learnt based on a set of training cases (x_i, y_i) with inputs x_i and targets y_i . Standard neural network training procedures adjust the weights and the biases in the network so as to minimize some error measure, most commonly the sum of squared deviations between the network outputs and the targets, that is $R(\theta) = \sum_i (O_i - y_i)^2$, where θ is the set of all network parameters. This amounts to maximum likelihood estimation when one adds an error term, say e_i , in the definition of O_i in Equation 1-4, and assumes normality of the errors. The error function $R(\theta)$ is nonconvex, possessing many local minima. As a result the final solution obtained is quite dependent on the choice of starting weights.

In the early nineties, Buntine and Weigand (1991) and Mackay (1992), showed that a principled Bayesian learning approach to neural networks can lead to many improvements. In particular, Mackay showed that by approximating the distributions of the weights with Gaussians and adopting smoothing priors, it is possible to

obtain estimates of the weights and output variances and to automatically set the regularization coefficients. Neal (1996) introduced advanced Bayesian simulation methods, specifically, the hybrid Monte Carlo method, into the analysis of neural networks.

Let us consider a typical network with one hidden layer represented by Equation 1-4. Here, θ represents the network parameters which consists of weights and bias units for input to hidden layer and hidden to output layer. The priors on the network parameters are given as follows.

Input to hidden units:

$$c_{js} \sim N(0, \sigma_u^2) \quad (1-5)$$

$$c_{j0} \sim N(0, \sigma_a^2), j = 1, \dots, M; s = 1, \dots, p.$$

Hidden to output units:

$$b_j \sim N(0, \sigma_v^2) \quad (1-6)$$

$$b_0 \sim N(0, \sigma_b^2), j = 1, \dots, M.$$

The variances can be kept fixed or we can put inverse gamma priors on them. In the Bayesian approach to neural network learning, the objective is to find the predictive distribution for the target values in a new "test" case (x_{n+1}, y_{n+1}) , given the inputs in that case, and the inputs and the targets in the training cases.

$$P(y_{n+1}|x_{n+1}, \text{Training Set}) = \int P(y_{n+1}|x_{n+1}, \theta) P(\theta|\text{Training Set}) d\theta \quad (1-7)$$

The posterior distribution for θ is typically very complex, with many modes. Finding the predictive distribution for a test case by evaluating the integral of Equation 1-7 is therefore a difficult task. A Markov chain Monte Carlo method based on the hybrid Monte Carlo algorithm is used to evaluate the integral. Hybrid Monte Carlo

algorithm is a form of the Metropolis algorithm in which the candidate states are found by means of dynamical simulation.

A new hierarchical Bayesian neural network is proposed by (Ghosh et al., 2004) for modeling binary data and thus can be used in classification context also. Consider random variables $y_1, \dots, y_n$ such that conditional on $p_1, \dots, p_n$, these are independent binary variables with success probabilities p_i , $i = 1, \dots, n$. In a specific application, p_i denotes the probability of the presence of certain characteristics. Model $\theta_i = \text{logit}(p_i)$ as shown in Equation 1-8.

$$\theta_i = \sum_{j=1}^M \beta_j \psi(\mathbf{x}_i^T \boldsymbol{\gamma}_j) + e_i = \boldsymbol{\beta}^T \boldsymbol{\eta}_i(\boldsymbol{\gamma}) + e_i, \quad (1-8)$$

where $\boldsymbol{\beta} = (\beta_1, \dots, \beta_M)^T$, $\mathbf{x}_i = (x_{i1}, \dots, x_{iM})^T$, $\boldsymbol{\eta}_i(\boldsymbol{\gamma}) = (\psi(\mathbf{x}_i^T \boldsymbol{\gamma}_1), \dots, \psi(\mathbf{x}_i^T \boldsymbol{\gamma}_M))^T$, $i = 1, \dots, n$. Also, the errors e_i are assumed to be iid $N(0, \sigma^2)$. The regression part of the model can be identified with the feedforward neural network. The $\mathbf{x}_i$ are the inputs, M is the number of hidden nodes (here assumed to be known), and β_j are the weights attached to the hidden nodes j ($j = 1, \dots, M$). The $\boldsymbol{\gamma}_j$ are the weights attached to the inputs for node j ($j = 1, \dots, M$), ψ is the activation function, and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)^T$.

The model given in Equation 1-8 should also be contrasted to a classical neural network model where the error components e_i are absent. Introduction of the e_i makes the present procedure more comparable to those based on generalized mixed linear models. Also, an MCMC implementation is greatly facilitated this way since most of the full conditionals then become standard distributions from which it is easy to generate samples.

At the first stage of the hierarchical prior, $\boldsymbol{\beta}, \boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_M$ are mutually independent with $\boldsymbol{\beta} \sim N(\mu_\beta \mathbf{1}_M, \sigma_\beta^2 \mathbf{I}_M)$ and $\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_M \stackrel{\text{iid}}{\sim} N(\boldsymbol{\mu}_\gamma, \mathbf{S}_\gamma)$. At the second stage, the prior parameters $\mu_\beta, \boldsymbol{\mu}_\gamma, \sigma^2, \sigma_\beta^2$ and $\mathbf{S}_\gamma$ are mutually independent with $\mu_\beta \sim N(a_\beta, A_\beta)$, $\boldsymbol{\mu}_\gamma \sim N_p(\mathbf{a}_\gamma, \mathbf{A}_\gamma)$, $\sigma^2 \sim \text{IG}(c_\sigma/2, c_\sigma C_\sigma/2)$, $\sigma_\beta^2 \sim \text{IG}(c_\beta/2, c_\beta C_\beta/2)$ and

$S_\gamma \sim \text{IW}(c_\gamma, c_\gamma^{-1} C_\gamma^{-1})$. Here IG and IW denote respectively the inverse gamma and inverse Wishart distributions. Specifically, σ_β^2 has pdf

$$f(\sigma_\beta^2) \propto \exp\left(-\frac{c_\beta C_\beta}{2\sigma_\beta^2}\right)(\sigma_\beta^2)^{-c_\beta/2-1}. \quad (1-9)$$

Also S_γ has pdf

$$f(S_\gamma) \propto |S_\gamma|^{-\frac{1}{2}(c_\gamma+p+1)} \exp\left[-\frac{1}{2}\text{tr}(S_\gamma^{-1} c_\gamma C_\gamma)\right]. \quad (1-10)$$

The selection of priors is usually problem-specific. Ideally, one would like to elicit these priors from past history. One interesting example is the *power prior* developed by Ibrahim et al. (1998) and Ibrahim and Chen (2000). However, for most practical applications, such information is unavailable. In such cases, it seems reasonable, at least from robustness considerations, to use near-diffuse priors for the hyperparameters of the hierarchical model.

The joint posterior of $\theta, \gamma, \beta, \mu_\beta, \mu_\gamma, \sigma^2, \sigma_\beta^2$ and S_γ given $\mathbf{y}$ is

$$\begin{aligned} & f(\theta, \gamma, \beta, \mu_\beta, \mu_\gamma, \sigma^2, \sigma_\beta^2, S_\gamma \mid \mathbf{y}) \\ & \propto \prod_{i=1}^N [\exp\{\phi_i^{-1}(\theta_i y_i - \kappa(\theta_i))\} (\sigma^2)^{-\frac{1}{2}N} \exp[-(2\sigma^2)^{-1} \sum_{i=1}^N (\theta_i - \eta_i^T(\gamma)\beta)^2] \\ & \times (\sigma_\beta^2)^{-\frac{1}{2}M} \exp[-\|\beta - \mu_\beta \mathbf{1}_M\|^2 / (2\sigma_\beta^2)] \\ & \times |S_\gamma|^{-\frac{1}{2}M} \exp[-\frac{1}{2} \sum_{j=1}^M (\gamma_j - \mu_\gamma)^T S_\gamma^{-1} (\gamma_j - \mu_\gamma)] \\ & \times \exp[-(2A_\beta)^{-1} (\mu_\beta - a_\beta)^2 - \frac{1}{2} (\mu_\gamma - \mathbf{a}_\gamma)^T \mathbf{A}_\gamma^{-1} (\mu_\gamma - \mathbf{a}_\gamma)] \\ & \times \exp[-(c_\sigma C_\sigma) / (2\sigma^2)] (\sigma^2)^{-\frac{c_\sigma}{2}-1} \\ & \times \exp[-(c_\beta C_\beta) / (2\sigma_\beta^2)] (\sigma_\beta^2)^{-\frac{c_\beta}{2}-1} \\ & \times |S_\gamma|^{-\frac{1}{2}(c_\gamma+p+1)} \exp[-\frac{1}{2}\text{tr}(S_\gamma^{-1} c_\gamma C_\gamma)]. \end{aligned} \quad (1-11)$$

One of the objectives is to find the posterior distribution of θ given $\mathbf{y}$, and use the same for finding the predictive distribution of the future unobserved $\mathbf{y}_u$ given the

observed $\mathbf{y}$. This is calculated by using Equation 1-12.

$$f(\mathbf{y}_u|\mathbf{y}) = \int f(\mathbf{y}_u|\boldsymbol{\theta})f(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}. \quad (1-12)$$

The posterior $f(\boldsymbol{\theta}|\mathbf{y})$, however, is analytically intractable as its evaluation requires high dimensional numerical integration. Thus Gibbs sampling is used to generate samples from this posterior. Some of the full conditionals needed in this procedure are available only up to unknown normalizing constants, and either a regular Metropolis-Hastings algorithm or the adaptive rejection sampling procedure of Gilks and Wild (1992) is used to sample from the conditionals.

Below we list these full conditionals:

- (i) $\beta|\boldsymbol{\theta}, \gamma, \mu_\beta, \boldsymbol{\mu}_\gamma, \sigma^2, \sigma_\beta^2, \mathbf{S}_\gamma, \mathbf{y}$
 $\sim N[(\sigma^{-2} \sum_{i=1}^N \boldsymbol{\eta}_i(\gamma) \boldsymbol{\eta}_i^T(\gamma) + \sigma_\beta^{-2} \mathbf{I}_M)^{-1} (\sigma^{-2} \sum_{i=1}^N \boldsymbol{\theta}_i \boldsymbol{\eta}_i(\gamma) + \sigma_\beta^{-2} \mu_\beta \mathbf{1}_M),$
 $(\sigma^{-2} \sum_{i=1}^N \boldsymbol{\eta}_i(\gamma) \boldsymbol{\eta}_i^T(\gamma) + \sigma_\beta^{-2} \mathbf{I}_M)^{-1}]$;
- (ii) $\mu_\beta|\boldsymbol{\theta}, \beta, \gamma, \mu_\gamma, \sigma^2, \sigma_\beta^2, \mathbf{S}_\gamma, \mathbf{y}$
 $\sim N[(M\sigma_\beta^{-2} + A_\beta^{-1})^{-1} (\sigma_\beta^{-2} \sum_{j=1}^M \beta_j + A_\beta^{-1} a_\beta), (M\sigma_\beta^{-2} + A_\beta^{-1})^{-1}]$;
- (iii) $\boldsymbol{\mu}_\gamma|\boldsymbol{\theta}, \beta, \gamma, \mu_\beta, \sigma^2, \sigma_\beta^2, \mathbf{S}_\gamma, \mathbf{y}$
 $\sim N_p[(M\mathbf{S}_\gamma^{-1} + \mathbf{A}_\gamma^{-1})^{-1} (\mathbf{S}_\gamma^{-1} \sum_{j=1}^M \gamma_j + \mathbf{A}_\gamma^{-1} \mathbf{a}_\gamma), (M\mathbf{S}_\gamma^{-1} + \mathbf{A}_\gamma^{-1})^{-1}]$;
- (iv) $\sigma^2|\boldsymbol{\theta}, \beta, \gamma, \mu_\beta, \boldsymbol{\mu}_\gamma, \sigma_\beta^2, \mathbf{S}_\gamma, \mathbf{y}$
 $\sim IG(\frac{N+c_\sigma}{2}, \frac{\sum_{i=1}^N (\boldsymbol{\theta}_i - \boldsymbol{\eta}_i^T(\gamma)\boldsymbol{\beta})^2 + c_\sigma C_\sigma}{2})$;
- (v) $\sigma_\beta^2|\boldsymbol{\theta}, \beta, \gamma, \mu_\beta, \boldsymbol{\mu}_\gamma, \sigma^2, \mathbf{S}_\gamma, \mathbf{y}$
 $\sim IG(\frac{M+c_\beta}{2}, \frac{\|\boldsymbol{\beta} - \mu_\beta \mathbf{1}_M\|^2 + c_\beta C_\beta}{2})$;
- (vi) $\mathbf{S}_\gamma|\boldsymbol{\theta}, \beta, \gamma, \mu_\beta, \boldsymbol{\mu}_\gamma, \sigma^2, \sigma_\beta^2, \mathbf{y}$
 $\sim IW(M + c_\gamma, (c_\gamma C_\gamma + \sum_{j=1}^M (\gamma_j - \boldsymbol{\mu}_\gamma)(\gamma_j - \boldsymbol{\mu}_\gamma)^T)^{-1})$;
- (vii) $\pi(\gamma|\boldsymbol{\theta}, \beta, \mu_\beta, \boldsymbol{\mu}_\gamma, \sigma^2, \sigma_\beta^2, \mathbf{S}_\gamma, \mathbf{y})$
 $\propto \exp[-\frac{1}{2} \{ \sum_{i=1}^N (\boldsymbol{\theta}_i - \boldsymbol{\eta}_i^T(\gamma)\boldsymbol{\beta})^2 + \sum_{j=1}^M (\gamma_j - \boldsymbol{\mu}_\gamma)^T \mathbf{S}_\gamma^{-1} (\gamma_j - \boldsymbol{\mu}_\gamma) \}]$;
- (viii) $\pi(\boldsymbol{\theta}_i|\boldsymbol{\theta}_j (j \neq i), \beta, \gamma, \mu_\beta, \boldsymbol{\mu}_\gamma, \sigma^2, \sigma_\beta^2, \mathbf{S}_\gamma, \mathbf{y})$
 $\propto \exp[\phi_i^{-1} \{ \boldsymbol{\theta}_i \boldsymbol{\theta}_i - \kappa(\boldsymbol{\theta}_i) \} - \frac{1}{2\sigma^2} (\boldsymbol{\theta}_i - \boldsymbol{\eta}_i^T(\gamma)\boldsymbol{\beta})^2]$.

The pdf's given in (i) through (vi) are standard, and it is easy to generate samples from the same. However, this is not so for the pdf's given in (vii) and (viii). Since the pdf given in (viii) is log-concave, so one can use the adaptive rejection sampling scheme of Gilks and Wild (1992). Generating samples from (vii) is even more complicated, and requires Metropolis-Hastings updates at every step. For efficient sampling, update γ in blocks (Roberts and Sahu, 1997). Also, to implement the Metropolis-Hastings algorithm, one needs a candidate generating density. Chib and Greenberg (1995) have several proposals in this regard. Since the conditional density of γ_j is proportional to $g_1(\gamma) \times g_2(\gamma_j)$, where $g_1(\gamma) = \exp[-(1/(2\sigma^2)) \sum_{i=1}^N (\theta_i - \beta_j \psi(\mathbf{x}_i^T \gamma_j) - \sum_{k \neq j} \beta_k \psi(\mathbf{x}_i^T \gamma_k))^2]$, which is bounded, and $g_2(\gamma_j) = \exp[-(1/2)(\gamma_j - \mu_\gamma)^T \mathbf{S}_\gamma^{-1} (\gamma_j - \mu_\gamma)]$, which is proportional to a multivariate normal pdf, following the third choice of Chib and Greenberg (1995), use $g_2(\gamma_j)$ as the proposal density to draw a new value, say, δ_j of γ_j with probability $\alpha_j = \min \left\{ 1, \frac{g_1(\delta_j)}{g_1(\gamma_j)} \right\}$.

The model of Ghosh et al. (2004) should be contrasted to Neal's (1996) model, since the latter does not account for the error components e_i . Also in their neural network model much more flexibility regarding the choice of priors and hyperpriors are incorporated, which greatly enhanced its learning performance.

1.4 Support Vector Machine

Support Vector Machines (SVM) are learning systems that use a hypothesis space of linear functions in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory. The SVM can be viewed in several contexts. In the classification context, they are motivated by the geometric interpretation of maximizing the margin of discrimination, and are characterized by the use of a kernel function. Support vector machine has its motivation from the optimal separating hyperplane theory.

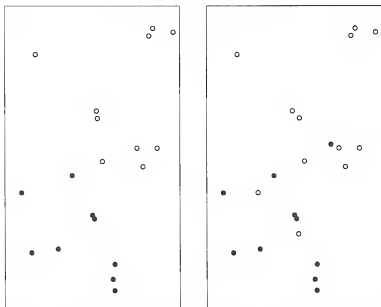


Figure 1-2: Separable and nonseparable cases for classification.

An optimal separating hyperplane separates the two classes and maximizes the distance to the closest point from either class (Vapnik, 1995). Let us consider a training set of n data points, $(\mathbf{x}_i, y_i)$, $i = 1, \dots, n$. Where $y_i \in \{-1, 1\}$, and $\mathbf{x}_i \in \mathbb{R}^p$. When the training data are separable, define a hyperplane by Equation 1-13.

$$\mathbf{x} : f(\mathbf{x}) = \beta_0 + \mathbf{x}^T \beta = 0 \quad (1-13)$$

with $\|\beta\| = 1$. The corresponding classification rule induced by $f(\mathbf{x})$ in Equation 1-14.

$$G(\mathbf{x}) = \text{sign}[f(\mathbf{x})] \quad (1-14)$$

To find the hyperplane with the biggest margin we have to solve the optimization problem in Equation 1-15.

$$\min_{\beta, \beta_0} \|\beta\| \quad \text{subject to } y_i(\beta_0 + \mathbf{x}_i^T \beta) \geq 1, i=1, \dots, n. \quad (1-15)$$

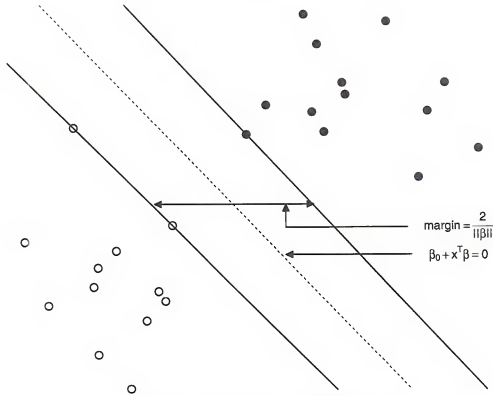


Figure 1-3: Support vector classifier.

This will give us an estimate of β_0 and β as $\hat{\beta}_0$ and $\hat{\beta}$ respectively. So for a new observation $\mathbf{x}_{n+1}$, the decision rule is shown in Equation 1-16.

$$\begin{aligned} G(\mathbf{x}_{n+1}) &= \text{sign}[\hat{\beta}_0 + \mathbf{x}^T \hat{\beta}] \\ &= \text{sign}[\hat{f}(\mathbf{x}_{n+1})] \end{aligned} \quad (1-16)$$

When the data is nonseparable (Figure 1-2), i.e the classes overlap, one way to deal is by adding slack variables $\xi = (\xi_1, \dots, \xi_n)$ and solving the following optimization problem in Equation 1-17.

$$\begin{aligned} \min \quad & \|\beta\| \\ \text{subject to } & y_i(\beta_0 + \mathbf{x}_i^T \beta) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i, \quad \sum \xi_i \leq C \text{ a constant.} \end{aligned} \quad (1-17)$$

An SVM maps the input vectors $\mathbf{x}$ into a high dimensional feature space through some nonlinear mapping, and then an optimal separating hyperplane is constructed in this high dimensional feature space. Then the solution function $f(\mathbf{x})$ can be written

as

$$\begin{aligned} f(\mathbf{x}) &= \beta_0 + \sum_{i=1}^n \alpha_i y_i < h(\mathbf{x}), h(\mathbf{x}_i) > \\ &= \beta_0 + \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) \end{aligned} \quad (1-18)$$

where $h(\mathbf{x}_i) = (h_1(\mathbf{x}_i), h_2(\mathbf{x}_i), \dots, h_m(\mathbf{x}_i))$ is the feature vector and $K(\mathbf{x}, \mathbf{x}_i) = < h(\mathbf{x}), h(\mathbf{x}_i) >$ is the kernel function. As the above expression can be expressed in kernel function we don't need to specify the transformation $h(\mathbf{x})$. The kernel function K should be positive definite. Two classical choices are (i) the Gaussian kernel $K(\mathbf{x}, \mathbf{x}_i) = \exp\{-\|\mathbf{x} - \mathbf{x}_i\|^2/\theta\}$ and (ii) the polynomial kernel $K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x} \cdot \mathbf{x}_i + 1)^\theta$. Here $\mathbf{a} \cdot \mathbf{b}$ denotes the inner product of the two vectors $\mathbf{a}$ and $\mathbf{b}$.

Wahba (1990, 1999) showed that SVM methodology can be cast as a regularization problem in terms of a reproducing kernel Hilbert space (RKHS) and hence SVMs and penalty methods, as used in the statistical theory of nonparametric regression, have strong a interrelationship. We have a training set $\{y_i, \mathbf{x}_i\}$, $i = 1, \dots, n$, where y_i is the response variable and $\mathbf{x}_i$ is the vector of covariate of size p corresponding to y_i . Given the training data our goal is to find an appropriate function $f(\mathbf{x})$ to predict the responses y in the test set based on the covariates $\mathbf{x}$. This can be viewed as a regularization problem of the form given in Equation 1-19.

$$\min_{f \in \mathcal{H}} \left[\sum_{i=1}^n L(y_i, f(\mathbf{x}_i)) + \lambda J(f) \right] \quad (1-19)$$

where $L(y, f(\mathbf{x}))$ is a loss function, $J(f)$ is a penalty functional, $\lambda > 0$ is the smoothing parameter, and $\mathcal{H}$ is a space of functions on which $J(f)$ is defined. A important subclass of problems of above form Equation 1-19 is generated by a positive definite kernel $K(\mathbf{x}, \mathbf{y})$ and the corresponding space of functions known as reproducing kernel Hilbert space (RKHS) denoted by H_K .

A Hilbert space is a vector space $\mathcal{H}$ with an inner product $\langle f, g \rangle$ such that the norm is defined by $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle}$. A formal definition of RKHS (Aronszajn, 1950; Parzen, 1970; Wahba, 1990) is as follows: Let T be a general index set. For example, $T = \{1, 2, \dots, N\}$, $T = [0, 1]$, or $T = E^d$, (Euclidean d-space), or $T = \mathcal{S}_d$ (the d-dimensional sphere). A kernel K in $\mathcal{H}$ is referred to as a reproducing kernel if $\langle K(t_1, \cdot), K(t_2, \cdot) \rangle = K(t_1, t_2)$ for every fixed $t_1 \in T$ and $t_2 \in T$. The space $\mathcal{H}$ is said to be a reproducing kernel Hilbert space (RKHS) with kernel K denoted as $\mathcal{H}_K$ if (i) for every fixed $t \in T$, $K(t, s) \in \mathcal{H}_K$, where $K(t, s)$ is considered as a function of s , and (ii) for every function f on T , $\langle f(s), K(t, s) \rangle = f(t)$ for every fixed t . By the Riesz representation theorem, for every RKHS $\mathcal{H}_K$ with kernel K , if $h \in \mathcal{H}$, then there exists an $M = M_t$ not depending on h such that $|h(t)| \leq M \|h\|_{\mathcal{H}}$. The familiar Hilbert space $L^2[0, 1]$ of square integrable functions on $[0, 1]$ does not have this property, as no such M exists.

A real symmetric function $K(s, t)$ is said to be positive definite on $T \times T$ if for every $n = 1, 2, \dots$, and every set of real numbers $\{a_1, a_2, \dots, a_n\}$ and $t_1, t_2, \dots, t_n, t_i \in T$, we have $\sum_{i,j=1}^n a_i a_j K(t_i, t_j) \geq 0$.

The penalty functional J can be defined in terms of kernels as well. For H_K the penalty functional is defined as $J(f) = \|f\|_{H_K}^2$. Rewriting Equation 1-19 we have,

$$\min_{f \in \mathcal{H}_K} \left[\sum_{i=1}^n L(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}_K}^2 \right] \quad (1-20)$$

It has been shown that the solution of Equation 1-20 is finite-dimensional (Wahba, 1990) and leads to a representation of f (Kimeldorf and Wahba, 1971; Wahba et al., 2002) as in Equation 1-21.

$$f_{\lambda}(x) = \sum_{i=1}^n \beta_i K(x, x_i) \quad (1-21)$$

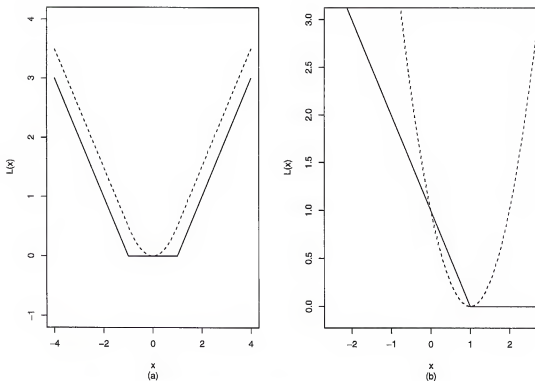


Figure 1-4: (a) Vapnik's ϵ -insensitive ($\epsilon = 1$). (b) Hinge loss.

It is also a property of RKHS that

$$\|h\|_{\mathcal{H}_K}^2 = \sum_{i,j=1}^n \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (1-22)$$

Representation of f in the form of Equation 1-21 is of special interest to us, because cases when the number of covariates p is much larger than the number of data points, one can effectively reduce the dimension of covariates from p to n . To obtain the estimate of f_λ substitute Equation 1-21 and Equation 1-22 in Equation 1-20 and then minimize it with respect to $\beta = (\beta_1, \dots, \beta_n)$ and the smoothing parameter λ . The other parameters inside the kernel K may be chosen by generalized approximate cross validation (GACV). An SVM can be fit into this regularization setup by a proper choice of loss functions as follows:

Choice of Loss functions (Figure 1-4). :

1. Hinge loss : $[1 - yf(\mathbf{x})]_+$, produces support vector classification.
2. Square error loss : $(y - f(\mathbf{x}))^2$, leads to least square support vector machine (LSVM).
3. Vapnik's ϵ -insensitive loss : $(|y - f(\mathbf{x})| - \epsilon)I(|y - f(\mathbf{x})| > \epsilon)$, produces support vector regression.

The classical support vector machine (SVM) algorithm, despite its success in regression and classification suffers from a serious disadvantage: it cannot provide any probabilistic outputs. Recently Tipping (2000) has formulated the relevance vector machine (RVM), a probabilistic model whose functional form is equivalent to the SVM. The original treatment of RVM relied on the use of type II maximum likelihood (Good, 1965) estimates of the hyper-parameters. Law and Kwok (2001) introduced a Bayesian formulation of SVM for regression, but they did not carry out a full Bayesian analysis and used instead certain approximations for the posterior. A similar remark applies to Sollich (2001) who considered SVM in the classification context. More recently, Mallick et al. (2005) considered a Markov chain Monte Carlo (MCMC) based full Bayesian analysis for classification where the number of covariates was far greater than the sample size as follows.

The hierarchical Bayesian SVM for classification is fomulated as

$$p(y_i|z_i) \propto \exp\{-L(y_i, z_i)\}, \quad i = 1, \dots, n, \quad (1-23)$$

where the $y_1, y_2, \dots, y_n$ are conditionally independent given latent variables $z_1, z_2, \dots, z_n$ and L is the hingeloss function explained earlier. Relate the latent variables z_i to $f(\mathbf{x}_i)$ by $z_i = f(\mathbf{x}_i) + \epsilon_i$, where the ϵ_i are residual random effects. This formulation is analogous to a generalized mixed linear model, but is slightly more general than the latter in that the likelihood is not necessarily restricted to the one-parameter exponential family.

As explained earlier, one can express f as in Equation 1-24.

$$f(\mathbf{x}_i) = \beta_0 + \sum_{j=1}^n \beta_j K(\mathbf{x}_i, \mathbf{x}_j | \boldsymbol{\theta}) \quad (1-24)$$

where K is a positive definite function of the covariates (inputs) $\mathbf{x}$ and one can allow some unknown parameters $\boldsymbol{\theta}$ to enrich the class of kernels.

The random latent variable z_i is thus modeled as in Equation 1-25.

$$z_i = \beta_0 + \sum_{j=1}^n \beta_j K(\mathbf{x}_i, \mathbf{x}_j | \boldsymbol{\theta}) + \epsilon_i = \mathbf{K}'_i \boldsymbol{\beta} + \epsilon_i, \quad (1-25)$$

where the ϵ_i are independent and identically distributed $N(0, \sigma^2)$ variables, and $\mathbf{K}'_i = (1, K(\mathbf{x}_i, \mathbf{x}_1 | \boldsymbol{\theta}), \dots, K(\mathbf{x}_i, \mathbf{x}_n | \boldsymbol{\theta}))$, $i = 1, \dots, n$.

To complete the hierarchical model, one needs to assign priors to the unknown parameters $\boldsymbol{\beta}$, $\boldsymbol{\theta}$ and σ^2 . Assign to $\boldsymbol{\beta}$ the Gaussian prior with mean $\mathbf{0}$ and variance $\sigma^2 \mathbf{D}_*^{-1}$, where $\mathbf{D}_* \equiv \text{Diag}(\lambda_1, \lambda, \dots, \lambda)$ is a $(n+1) \times (n+1)$ diagonal matrix, λ_1 being fixed at a small value, but λ is unknown. This amounts to a large variance for the intercept term. Assign a proper uniform prior to $\boldsymbol{\theta}$, an inverse Gamma prior to σ^2 and a Gamma prior to λ . A $\text{Gamma}(\alpha, \xi)$ distribution for a random variable, say U , has probability density function proportional to $\exp(-\xi u) u^{\alpha-1}$, while the reciprocal of U will then be said to have a $\text{IG}(\alpha, \xi)$ distribution. Our model is thus given by Equation 1-26.

$$\begin{aligned} p(y_i | z_i) &\propto \exp\{-L(y_i, z_i)\}; \\ z_i | \boldsymbol{\beta}, \boldsymbol{\theta}, \sigma^2 &\stackrel{\text{ind}}{\sim} N_1(z_i | \mathbf{K}'_i \boldsymbol{\beta}, \sigma^2); \\ \boldsymbol{\beta}, \sigma^2 &\sim N_{n+1}(\boldsymbol{\beta} | \mathbf{0}, \sigma^2 \mathbf{D}_*^{-1}) \text{IG}(\sigma^2 | \gamma_1, \gamma_2), \\ \boldsymbol{\theta} &\sim \prod_{q=1}^p U(a_{q1}, a_{q2}) \\ \lambda &\sim \text{Gamma}(m, c) \end{aligned} \quad (1-26)$$

where $U(a_{q1}, a_{q2})$ is the uniform probability density function over (a_{q1}, a_{q2}) .

This Bayesian model is similar to the RKHS model as the precision parameter or the ridge parameter λ is similar to the smoothing parameter and the exponent of the Gaussian prior for β is essentially equivalent to the quadratic penalty function in RKHS context.

This model can also be extended using multiple smoothing parameters so that the prior for (β, σ^2) is given by Equation 1-27.

$$\beta, \sigma^2 \sim N_{n+1}(\beta|0, \sigma^2 \mathbf{D}^{-1}) \text{IG}(\sigma^2|\gamma_1, \gamma_2), \quad (1-27)$$

where $\mathbf{D}$ is a diagonal matrix with diagonal elements $\lambda_1, \dots, \lambda_{n+1}$. Once again λ_1 is fixed at a small value, but all other λ 's are unknown. Assign independent $\text{Gamma}(m, c)$ priors to them. Let $\lambda = (\lambda_1, \dots, \lambda_{n+1})'$.

In this Bayesian formulation, this optimization problem of classical SVM is equivalent to finding the posterior mode of β , where the likelihood is given by $\exp[-\sum_{i=1}^n \{1 - y_i f(\mathbf{x}_i)\}_+]$, while β has the $N(0, C\mathbf{I}_{n+1})$ prior. Also instead of a point estimate one will be able to obtain the full posterior predictive distribution of a new observation.

1.5 Multiple Linear Regression

The general purpose of multiple regression (the term was first used by Pearson, 1908) is to learn more about the relationship between several independent or predictor variables and a dependent or criterion variable (Hastie et al., 2001, page 54–55). In general then, multiple regression procedures will estimate a linear equation of the form given in Equation 1-28.

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots + \beta_p X_{ip} + \epsilon_i \\ \mathbf{Y} &= \mathbf{X}\beta + \epsilon \end{aligned} \quad (1-28)$$

where $\mathbf{X} = (\mathbf{1}_n^T, X_{ij})$, $i = 1, \dots, n$, $j = 1, \dots, p$, $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_p)^T$, and $\mathbf{Y} = (Y_1, \dots, Y_n)^T$. The regression line expresses the best prediction of the dependent variable (Y), given the independent variables (X). We use least square estimate to fit a multiple linear regression. Hence the estimated $\boldsymbol{\beta}$ is given by Equation 1-29.

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}. \quad (1-29)$$

There are several drawbacks of MLR. First of all, as is evident in the name multiple linear regression, it is assumed that the relationship between variables is linear. In practice this assumption can virtually never be confirmed; fortunately, multiple regression procedures are not greatly affected by minor deviations from this assumption. Secondly, choice of variables which are to be included in the model is another tough problem. And the problem is compounded when, in addition, the number of observations is relatively low. Thirdly, when the independent variables are highly correlated, it is impossible to come up with reliable estimates of their individual regression coefficients. This is because in this situation $(\mathbf{X}^T \mathbf{X})^{-1}$ cannot be calculated. This is known as the problem of multicollinearity.

1.6 Principal Component Regression

PCR, or principal component regression, is a simple extension of MLR and PCA. In the first step, the principal components are calculated. The scores of the most important principal components are used as the basis for the multiple linear regression with the target data $\mathbf{Y}$, (Hastie et al., 2001, page 66-67). The purpose of principal component regression (PCR) is to estimate the values of a response variable at the basis of selected principal components (PCs) of the explanatory variables. There are two main reasons for regressing the response variable on the PCs rather than directly on the explanatory variables. Firstly, the explanatory variables are often highly correlated (multicollinearity) which may cause inaccurate estimations of the least squares (LS) regression coefficients. This can be avoided by using the PCs in place

of the original variables since the PCs are uncorrelated. Secondly, the dimensionality of the regressors is reduced by taking only a subset of PCs for prediction.

$$y^{PCR} = \bar{Y} + \sum_{r=1}^M \hat{\theta}_r \mathbf{z}_r \quad (1-30)$$

where $\mathbf{z}_r = \mathbf{X} v_m$ is the factor scores, M is the number of principal components included in the model, and $\hat{\theta}_r = \frac{\langle \mathbf{z}_r, \mathbf{Y} \rangle}{\langle \mathbf{z}_r, \mathbf{z}_r \rangle}$.

The intention, in using PCR, was to extract the underlying effects in the $\mathbf{X}$ data, and to use these to predict the $\mathbf{Y}$ values. In this way, we could guarantee that only independent effects were used, and that low-variance noise effects were excluded. This improved the quality of the model significantly. However, PCR still has a problem: if the relevant underlying effects are small in comparison with some irrelevant ones, then they may not appear among the first few principal components. So, we are left with a component selection problem - we cannot just include the first n principal components, as these may serve to degrade the performance of the model. Instead, we have to extract all components, and determine whether adding each one of these, improves the model or not. This is a complex problem.

1.7 Partial Least Squares

Partial least squares regression (Hastie et al., 2001, page 66–68) is an extension of the multiple linear regression model. Partial least squares regression has been used in various disciplines such as chemistry, economics, medicine, psychology, and pharmaceutical science where predictive linear modeling, especially with a large number of predictors, is necessary. Especially in chemometrics, partial least squares regression has become a standard tool for modeling linear relations between multivariate measurements.

As in multiple linear regression, the main purpose of partial least squares regression is to build a linear model,

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E} \quad (1-31)$$

where $\mathbf{Y}$ is an n cases by m variables response matrix, $\mathbf{X}$ is an n cases by p variables predictor (design) matrix, $\mathbf{B}$ is a p by m regression coefficient matrix, and $\mathbf{E}$ is a noise term for the model which has the same dimension as $\mathbf{Y}$. Usually, the variables in $\mathbf{X}$ and $\mathbf{Y}$ are centered by subtracting their means and scaled by dividing by their standard deviations.

Like principal components regression, partial least squares regression produce factor scores as linear combinations of the original predictor variables so that there is no correlation between the factor score variables used in the predictive regression model. For example, suppose we have a data set with response variables $\mathbf{Y}$ (in matrix form) and a large number of predictor variables $\mathbf{X}$ (in matrix form), some of which are highly correlated. A regression using factor extraction for this type of data computes the factor score matrix $\mathbf{T} = \mathbf{XW}$ for an appropriate weight matrix $\mathbf{W}$, and then considers the linear regression model $\mathbf{Y} = \mathbf{TQ} + \mathbf{E}$, where $\mathbf{Q}$ is a matrix of regression coefficients (loadings) for $\mathbf{T}$, and $\mathbf{E}$ is an error (noise) term. Once the loadings $\mathbf{Q}$ are computed, the above regression model is equivalent to $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, where $\mathbf{B} = \mathbf{WQ}$, which can be used as a predictive regression model.

Partial least squares regression differs from principal components regression in the methods used in extracting factor scores. In short, partial least squares regression produces the weight matrix $\mathbf{W}$ reflecting the covariance structure between the predictor and response variables.

For establishing the model, partial least squares regression produces a p by c weight matrix $\mathbf{W}$ for $\mathbf{X}$ such that $\mathbf{T} = \mathbf{XW}$, i.e., the columns of $\mathbf{W}$ are weight vectors for the $\mathbf{X}$ columns producing the corresponding n by c factor score matrix $\mathbf{T}$.

These weights are computed so that each one of them maximizes the covariance between responses and the corresponding factor scores. Ordinary least squares procedures for the regression of $\mathbf{Y}$ on $\mathbf{T}$ are then performed to produce $\mathbf{Q}$, the loadings for $\mathbf{Y}$ (or weights for $\mathbf{Y}$) such that $\mathbf{Y} = \mathbf{TQ} + \mathbf{E}$. Once $\mathbf{Q}$ is computed, we have $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, where $\mathbf{B} = \mathbf{WQ}$, and the prediction model is complete.

One additional matrix which is necessary for a complete description of partial least squares regression procedures is the p by c factor loading matrix $\mathbf{P}$ which gives a factor model $\mathbf{X} = \mathbf{TP} + \mathbf{F}$, where $\mathbf{F}$ is the unexplained part of the $\mathbf{X}$ scores. The standard algorithms for computing partial least squares regression components (i.e., factors) are nonlinear iterative partial least squares (NIPALS) (Wold, 1966) and SIMPLS algorithm (Jong, 1993).

In short, partial least squares regression is probably the least restrictive of the various multivariate extensions of the multiple linear regression model. This flexibility allows it to be used in situations where the use of traditional multivariate methods is severely limited, such as when there are fewer observations than predictor variables.

1.8 Random Forest

Random forests introduced by Breiman (2001) are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Random forest has been very successfully used for both classification as well as regression purposes.

The random forest classifier uses a large number of individual decision trees and decides the class by choosing the mode or the most frequently occurring of the classes as determined by the individual trees.

The individual trees are constructed by the following steps:

- Step 1. Assume that the number of cases in the training set is n , and that the number of variables in the classifier is p . Select the number of input variables that

will be used to determine the decision at a node of the tree. This number, m should be much much less than p .

- Step 2. Choose a training set by choosing n samples from the training set with replacement.
- Step 3. For each node of the tree randomly select m of the p variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.
- Step 4. Each tree is fully grown and not pruned, as would be done in constructing a normal tree classifier).
- Step 5. After each tree is built, all of the data from test set are run down the trees, and final class prediction is obtained by a majority voting rule.

The random forest classifier has several advantages and considered to be the most accurate “out of shelf” algorithm for classification among current algorithms (as of 2005). It can handle a very large number input variables and can estimate the importance of variables in determining classification. It generates an internal unbiased estimate of the generalization error as the forest building progresses. Also it includes a good method for estimating missing data and maintains accuracy when a large proportion of the data is missing and provides an experimental method for detecting variable interactions. Random forest avoids the problem of overfitting and no separate test set is required to asses its performance.

Similar kind of implementation can be devised for complicated regression problems with regression trees with numerical output variables rather than class labels. The final prediction is obtained by averaging over all trees in the forest.

1.9 Motivation and Outline of the Study

In this dissertation we propose three different methods extending the ideas discussed in Sections 1.4 and 1.5 in the Bayesian paradigm.

In Chapter 2 we introduce Hierarchical Bayesian neural network (HBNN) for modeling bivariate binary data. We have applied our model on a number of data sets and compare their performance against the existing machineries. Our main application is in classification problems for prostate cancer data, where we try to predict certain features indicative of non-organ confined prostate cancer on the basis of the results of certain diagnostic tests administered to patients using our proposed hierarchical Bayesian neural network model. This chapter also gives details for the implementation of the MCMC algorithm and gives a comparative performance of our proposed method with other standard tools. In total, two real life data sets and one simulated data set are used for application.

In Chapter 3 we develop a Bayesian support vector and relevance vector regression model for large p small n or “high dimension and low sample size” problems. We have extended the full Bayesian support vector regression (SVR) and relevance vector regression (RVR) models when the response is multivariate. This method is applied to a high dimensional data set of near infra-red spectroscopy for detecting the composition of biscuit dough, and a second data set arising from research in fluorescence based optics for continuously monitoring glucose levels in the body of diabetics.

In Chapter 4 we propose a full probabilistic model-based approach, specifically probabilistic relevance vector machine (RVM), as well as support vector machines (SVM) for multicategory classification. A hierarchical model is also proposed where the unknown smoothing parameter is interpreted as a shrinkage parameter. We assign a prior distribution to it and obtain its posterior distribution using Bayesian methods. In this way, both the point predictions as well as the associated measures of uncertainty is obtained. We also propose a Bayesian variable selection method for selecting the differentially expressed genes, integrated with the RVM and SVM models for improved classification. This method makes use of mixture priors and

Markov chain Monte Carlo technique to identify the important predictors (genes) and classify samples simultaneously. All the models are used to tackle the problem of cancer classification in the context of multiple-tumor types. We have applied our method to two different microarray datasets in order to identify differentially expressed genes and compared their classification accuracy with the existing methods.

In Chapter 5 we summarize all the developed methods and suggest some new directions for further investigation.

CHAPTER 2

BAYESIAN NEURAL NETWORK FOR BIVARIATE BINARY DATA AND ITS APPLICATION TO PROSTATE CANCER STUDY

“The human brain, then, is the most complicated organization of matter that we know.” —Isaac Asimov

This chapter presents a new way of predicting certain features indicative of non-organ confined prostate cancer, using the state of the art neural network technique, on the basis of certain diagnostic tests administered to patients suffering from prostate cancer. All previous attempts were confined to the prediction of a single feature of the non-organ confined prostate cancer at a time. In this chapter, we make an attempt to predict jointly several features indicative of non-organ confined cancer. Ghosh et al. (2004), used Bayesian neural network for prediction. But their method is based on the prediction of a single feature at a time. We introduce a bivariate hierarchical Bayesian neural network which provides the posterior (or predictive) probability of presence or absence of these features jointly in a patient, based on certain diagnostic tests. The recent advent of DNA microarray technique has made simultaneous monitoring of thousands of gene expression possible. Prediction of features indicative of non organ confined cancer, based on gene expression profile may provide more information than the conventional clinical or pathological information. We have also applied our model to predict the features in a patient based on gene expression microarrays. This method can be used for modeling any type of bivariate binary data. In this chapter our sole focus will be its application to the prostate cancer study.

In Section 2.1 of this chapter, we shall discuss the prostate cancer disease in American men and the various features which are indicative of non-organ confined prostate cancer. In Section 2.2, we will briefly review standard methods like bivariate

logit regression, feedforward neural network of Ripley (1996) and a Bayesian approach proposed by Neal (1996). In Section 2.3, we introduce our hierarchical Bayesian neural network models for bivariate binary data. This method extends the neural network methodology introduced by Ghosh et al. (2004) to multivariate binary data. Here we will also discuss the implementation of our methodology via Markov chain Monte Carlo (MCMC). In Section 2.4, we apply our methodology developed in Section 2.3 to the prostate cancer problem and compare the same with the other standard methods discussed in Section 2.2. Three different data sets are considered for this purpose. The first one is a prostate cancer dataset with clinical covariates, the second one is another prostate cancer study with gene expression microarrays, while the third one is a simulated dataset. In Section 2.5 we make some concluding remarks, and suggest some future possibilities.

2.1 Prostate Cancer

Prostate cancer develops from the growth of cancerous cells within the prostate gland. It is also the second leading cause of cancer death in men, exceeded only by lung cancer. It is the most common type of cancers found in American men, other than skin cancer. The American Cancer Society estimates that there will be about 230,900 new cases of prostate cancer in the United States in the year 2004. About 29,900 men will die of this disease. It is empirically found that every 1 man in 6 will get prostate cancer during his lifetime, and every 1 man in 32 will die of this disease. Beyond the human costs of prostate cancer, the financial burden imposed by the disease is staggering. The National Cancer Institute (NCI) estimates that \$ 5 billion is spent annually in direct medical care costs.

The prostate cancer may be locally confined or it may spread outside the organ. When locally confined, there are several options for treating and curing this disease; otherwise surgery is the only option. Hence, it is very important to know based on pre-surgery biopsy, how likely the cancer is, organ confined or not. Features that

are indicators of non organ confined cancer are - margin positivity (MP) and seminal vesicle (SV) positivity. As surgery is the only option for the cancer spread outside the organ, it is very important to identify accurately the stage of the disease. In the next two sections, we will discuss several methods for predicting jointly the probabilities of presence of margin positivity and seminal vesicle positivity in a patient. Based on these probabilities, the doctors can make a decision for whether or not to go for surgical intervention.

2.2 Standard Methods

There are several methods available for modeling bivariate binary data in this section. We will briefly discuss three of them. They are the bivariate logit model, Ripley's feedforward neural network and Neal's Bayesian neural network.

2.2.1 Bivariate Logit Model

We write $\mathbf{Y} = (Y_1, Y_2)^T$, where Y_1 and Y_2 each takes only the values 0 and 1. In this example, $\mathbf{Y}$ is a bivariate random vector where Y_1 takes the value 1 or 0 according to the presence or absence of MP, and Y_2 takes the value 1 or 0 according to the presence or absence of SV. Let $p_{rs} = P(Y_1 = r, Y_2 = s)$, $r, s = 0, 1$, be the joint probabilities, and $p_j = P(Y_j = 1)$, $j = 1, 2$, be the marginal probabilities. The bivariate logistic model (BLM) described in section 6.5.6 of McCullagh and Nelder (1989) and Palmgren (1989) is specified by modelling the marginal distributions of each Y_j , and the odds ratio, $\psi = (p_{00}p_{11})/(p_{10}p_{01})$. The model is given by Equation 2-1.

$$\begin{aligned}\text{logit}(p_j) &= \eta_j(\mathbf{x}), \quad j = 1, 2; \\ \log(\psi(\mathbf{x})) &= \eta_3(\mathbf{x}),\end{aligned}\tag{2-1}$$

where $\eta_j = \beta_j^T x$, $j = 1, 2, 3$, and x is the predictor variable or the covariates. The probability p_{11} can be obtained from p_1 , p_2 and ψ as in Equation 2-2.

$$p_{11} = \begin{cases} \frac{1}{2}(\psi - 1)^{-1}\{a - \sqrt{a^2 + b}\} & \text{if } \psi \neq 1 \\ p_1 p_2 & \text{if } \psi = 1, \end{cases} \quad (2-2)$$

where $a = 1 + (p_1 + p_2)(\psi - 1)$ and $b = -4\psi(\psi - 1)p_1 p_2$, Dale (1986). The other three joint probabilities p_{rs} can then be recovered easily from the marginals and p_{11} . The above bivariate logistic model is fitted using the iteratively reweighted least squares technique, and the details are discussed in section 6.5.6 of McCullagh and Nelder (1989) and section 4.6.3 of Agresti (2002).

The BLM is similar to the bivariate probit model but has several advantages: it is computationally simpler, and odds ratios are preferred to correlation coefficients when describing the association between two binary variables. However in theory, there is no reason why other link functions could not be used for marginal probabilities.

2.2.2 Neural Networks

Neural Networks as proposed by Ripley (1996) and Neal (1996) (see Section 1.4) can be used to model the bivariate binary data. Its highly nonlinear structure makes it a very powerful tool for predicting the stages of prostate cancer. But they remain handicapped due their inability to incorporate the dependency among the two components of a bivariate binary data. That means to fit a bivariate binary data, we just fit separate neural networks to the two different components separately without considering any possible association among the components. This can often cause significant efficiency loss for inferential purposes.

2.3 Hierarchical Bayesian Neural Network Model for Bivariate Binary Data

Let $y_1, y_2, \dots, y_n$ be independent random vectors, where $y_i = (y_{i1}, y_{i2})^T$ and y_{ik} can take values 0 or 1, $i = 1, \dots, n$, $k = 1, 2$.

Let

$$\begin{aligned} y_{i1}|p_{i1} &\stackrel{iid}{\sim} \text{Bin}(1, p_{i1}); \\ y_{i2}|p_{i2} &\stackrel{iid}{\sim} \text{Bin}(1, p_{i2}). \end{aligned} \quad (2-3)$$

Assume that conditional on p_{i1} and p_{i2} , y_{i1} and y_{i2} are independent. Define

$$\begin{aligned} \theta_{ik} &= \text{logit}(p_{ik}) \\ &= \text{log}(p_{ik}/(1 - p_{ik})), \end{aligned} \quad (2-4)$$

where $i = 1, 2, \dots, n; k = 1, 2$. Now model θ_{ik} using a single hidden layer feedforward Neural Network as in Equation 2-5.

$$\theta_{ik} = \sum_{j=1}^M \beta_{jk} \psi(\mathbf{x}_i^T \boldsymbol{\gamma}_{jk}) + e_{ik}, \quad (2-5)$$

where $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ is the input vector or covariates corresponding to $\mathbf{y}_i$, M is the number of nodes in the hidden layer of the neural network, and ψ is the activation function. Consider M as fixed and take ψ as the logistic function. Also, assume

$$\mathbf{e}_i = (e_{i1}, e_{i2})^T \stackrel{iid}{\sim} N_2(\mathbf{0}, \boldsymbol{\Sigma}_e). \quad (2-6)$$

This introduction of $\mathbf{e}_i$ enables us to account for any unexplained sources of variation not captured by the model. Although conditional on p_{i1} and p_{i2} , y_{i1} and y_{i2} are independent, we establish the dependency between them by introducing the correlated error $\mathbf{e}_i$. Rewriting Equation 2-5 we have Equation 2-7.

$$\boldsymbol{\theta}_i = \boldsymbol{\beta}^T \boldsymbol{\eta}_i(\boldsymbol{\gamma}) + \mathbf{e}_i, \quad (2-7)$$

where $\boldsymbol{\theta}_i = (\theta_{i1}, \theta_{i2})^T$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$, $\boldsymbol{\beta}_k = (\beta_{1k}, \dots, \beta_{Mk})^T$, $\boldsymbol{\eta}_i = (\eta_{i1}(\boldsymbol{\gamma}), \eta_{i2}(\boldsymbol{\gamma}))$, and $\boldsymbol{\eta}_{ik} = (\psi(\mathbf{x}_i^T \boldsymbol{\gamma}_{1k}), \dots, \psi(\mathbf{x}_i^T \boldsymbol{\gamma}_{Mk}))^T$, $i = 1, \dots, n$, $k = 1, 2$.

We assign mutually independent priors on the network parameters β and γ as well as on Σ_e the error covariance matrix described in Equation 2-8.

$$\begin{aligned}\beta_k &\stackrel{iid}{\sim} N_M(\mu_{\beta_k} \mathbf{1}_M, \sigma_{\beta_k}^2 \mathbf{I}_M); \\ \gamma_{jk} &\stackrel{iid}{\sim} N_p(\mu_{\gamma_k}, \mathbf{S}_{\gamma_k}); \\ \Sigma_e &\sim \text{IW}(c_e, c_e^{-1} \mathbf{C}_e^{-1}),\end{aligned}\tag{2-8}$$

where $j = 1, \dots, M$ and $k = 1, 2$. At the second stage we put mutually independent priors on first stage prior parameters as in Equation 2-9.

$$\begin{aligned}\mu_{\beta_k} &\sim N(a_{\beta_k}, A_{\beta_k}) \\ \sigma_{\beta_k}^2 &\sim \text{IG}(c_{\beta_k}/2, c_{\beta_k} C_{\beta_k}/2) \\ \mu_{\gamma_k} &\sim N_p(\mathbf{a}_{\gamma_k}, \mathbf{A}_{\gamma_k}) \\ \mathbf{S}_{\gamma_k} &\sim \text{IW}(c_{\gamma_k}, c_{\gamma_k}^{-1} \mathbf{C}_{\gamma_k}^{-1}),\end{aligned}\tag{2-9}$$

where $k=1,2$. Here IG and IW denote respectively the inverse gamma and inverse Wishart distributions. Specifically, $\sigma_{\beta_k}^2$ has pdf

$$f(\sigma_{\beta_k}^2) \propto \exp\left(-\frac{c_{\beta_k} C_{\beta_k}}{2\sigma_{\beta_k}^2}\right) (\sigma_{\beta_k}^2)^{-c_{\beta_k}/2-1}.\tag{2-10}$$

Also $\mathbf{S}_{\gamma_k}$ has pdf

$$f(\mathbf{S}_{\gamma_k}) \propto |\mathbf{S}_{\gamma_k}|^{-\frac{1}{2}(c_{\gamma_k}+p+1)} \exp\left[-\frac{1}{2} \text{tr}(\mathbf{S}_{\gamma_k}^{-1} c_{\gamma_k} \mathbf{C}_{\gamma_k})\right].\tag{2-11}$$

The priors for the network parameters is to embody our prior belief about the model. But in most practical applications we have very little idea about it. So here we chose near-diffuse priors for the hyperparameters of the model. A hierarchical model, in general, has its virtues. It is richer than a one-stage model, and produces automatically shrinkage estimators of parameters. These shrinkage estimators, by borrowing strength, usually perform better than the regular estimators.

The joint posterior of $\theta, \gamma_1, \gamma_2, \beta, \mu_{\beta_1}, \mu_{\beta_2}, \sigma_{\beta_1}^2, \sigma_{\beta_2}^2, \Sigma_e, S_{\gamma_1},$ and S_{γ_2} given $\mathbf{y}$ is

$$\begin{aligned}
& f(\theta, \gamma_1, \gamma_2, \beta, \mu_{\beta_1}, \mu_{\beta_2}, \sigma_{\beta_1}^2, \sigma_{\beta_2}^2, \Sigma_e, S_{\gamma_1}, S_{\gamma_2} | \mathbf{y}) \\
& \propto \prod_{i=1}^n [p_{i1}^{y_{i1}} (1 - p_{i1})^{1-y_{i1}} p_{i2}^{y_{i2}} (1 - p_{i2})^{1-y_{i2}}] \\
& \times |\Sigma_e|^{-\frac{n}{2}} \exp \left\{ -\frac{1}{2} \sum_{i=1}^n \{ \theta_i - \beta^T \eta_i(\gamma) \}^T \Sigma_e^{-1} \{ \theta_i - \beta^T \eta_i(\gamma) \} \right\} \\
& \times \prod_{k=1}^2 \left[(\sigma_{\beta_k}^2)^{-\frac{1}{2}M} \exp \left\{ -\|\beta_k - \mu_{\beta_k} \mathbf{1}_M\|^2 / (2\sigma_{\beta_k}^2) \right\} \right] \\
& \times \prod_{k=1}^2 \left[|S_{\gamma_k}|^{-\frac{1}{2}M} \exp \left\{ -\frac{1}{2} \sum_{j=1}^M (\gamma_{jk} - \mu_{\gamma_k})^T S_{\gamma_k}^{-1} (\gamma_{jk} - \mu_{\gamma_k}) \right\} \right] \quad (2-12) \\
& \times \prod_{k=1}^2 \left[\exp \left\{ -(2A_{\beta_k})^{-1} (\mu_{\beta_k} - a_{\beta_k})^2 - \frac{1}{2} (\mu_{\gamma_k} - a_{\gamma_k})^T A_{\gamma_k}^{-1} (\mu_{\gamma_k} - a_{\gamma_k}) \right\} \right] \\
& \times \prod_{k=1}^2 \left[\exp \left\{ -(c_{\beta_k} C_{\beta_k}) / (2\sigma_{\beta_k}^2) \right\} (\sigma_{\beta_k}^2)^{-\frac{c_{\beta_k}}{2}-1} \right] \\
& \times \prod_{k=1}^2 \left[|S_{\gamma_k}|^{-\frac{1}{2}(c_{\gamma_k} + p + 1)} \exp \left\{ -\frac{1}{2} \text{tr}(S_{\gamma_k}^{-1} c_{\gamma_k} C_{\gamma_k}) \right\} \right] \\
& \times |\Sigma_e|^{-\frac{1}{2}(c_e + 3)} \exp \left\{ -\frac{1}{2} \text{tr}(\Sigma_e^{-1} c_e C_e) \right\}.
\end{aligned}$$

The Bayesian inference of the network parameters is based on the joint posterior distribution in Equation 2-12. Our aim is to estimate this joint distribution from which, by standard probability marginalization and transformation techniques, one can “theoretically” obtain all posterior features of interest. For instance to predict the value of a new unknown observation $\mathbf{y}_u$ we need to integrate the predictions of the model with respect to the posterior distribution of the parameters to get the posterior predictive density, given by Equation 2-13.

$$f(\mathbf{y}_u | \mathbf{y}) = \int f(\mathbf{y}_u | \Theta) f(\Theta | \mathbf{y}) d\Theta \quad (2-13)$$

where Θ stands for all parameters of the neural network model. However, it is not possible to obtain these quantities analytically, as it requires the evaluation of high dimensional integrals of nonlinear functions in the parameters. Thus we propose an alternative approach based on Gibbs sampler Gelfand and Smith (1990) to perform the Bayesian computation. It requires generating samples from the full conditionals of the different parameters given the remaining parameters and the data. The key idea is to build an ergodic Markov chain, whose equilibrium distribution is the desired posterior distribution. We list the full conditionals as follows:

- (i) $\pi(\beta | (\dots)) \sim \exp[-\frac{1}{2} \sum_{i=1}^n (\theta_i - \beta^T \eta_i(\gamma))^T \Sigma_e^{-1} (\theta_i - \beta^T \eta_i(\gamma))] \times \prod_{k=1}^2 \left[(\sigma_{\beta_k}^2)^{-\frac{1}{2}M} \exp \left\{ -\|\beta_k - \mu_{\beta_k} \mathbf{1}_M\|^2 / (2\sigma_{\beta_k}^2) \right\} \right];$
- (ii) $\mu_{\beta_k} | (\dots) \sim N[(M\sigma_{\beta_k}^{-2} + A_{\beta_k}^{-1})^{-1}(\sigma_{\beta_k}^{-2}\beta_k^T \mathcal{C}_k + A_{\beta_k}^{-1}a_{\beta_k}), (M\sigma_{\beta_k}^{-2} + A_{\beta_k}^{-1})^{-1}];$
- (iii) $\mu_{\gamma_k} | (\dots) \sim N_p[(M\mathcal{S}_{\gamma_k}^{-1} + A_{\gamma_k}^{-1})^{-1}(\mathcal{S}_{\gamma_k}^{-1} \sum_{j=1}^M \gamma_{jk} + A_{\gamma_k}^{-1}a_{\gamma_k}), (M\mathcal{S}_{\gamma_k}^{-1} + A_{\gamma_k}^{-1})^{-1}];$
- (iv) $\Sigma_e | (\dots) \sim \text{IW}(n + c_e, \{c_e \mathcal{C}_e + \sum_{i=1}^n (\theta_i - \beta^T \eta_i(\gamma))(\theta_i - \beta^T \eta_i(\gamma))^T\}^{-1});$
- (v) $\sigma_{\beta_k}^2 | (\dots) \sim \text{IG}(\frac{M+c_{\beta_k}}{2}, \frac{\|\beta_k - \mu_{\beta_k} \mathbf{1}_M\|^2 + c_{\beta_k} \mathcal{C}_{\beta_k}}{2});$
- (vi) $\mathcal{S}_{\gamma_k} | (\dots) \sim \text{IW}(M + c_{\gamma_k}, \{c_{\gamma_k} \mathcal{C}_{\gamma_k} + \sum_{j=1}^M (\gamma_{jk} - \mu_{\gamma_k})(\gamma_{jk} - \mu_{\gamma_k})^T\}^{-1});$
- (vii) $\pi(\gamma | (\dots)) \propto \prod_{k=1}^2 [\exp\{-\frac{1}{2} \{ \sum_{j=1}^M (\gamma_{jk} - \mu_{\gamma_k})^T \mathcal{S}_{\gamma_k}^{-1} (\gamma_{jk} - \mu_{\gamma_k}) + \sum_{i=1}^n (\theta_i - \beta^T \eta_i(\gamma))^T \Sigma_e^{-1} (\theta_i - \beta^T \eta_i(\gamma)) \} \}], \text{ where } \gamma_j = (\gamma_{j1}^T, \gamma_{j2}^T)^T;$
- (viii) $\pi(\theta_i | \theta_j (j \neq i), (\dots)) \propto \prod_{i=1}^n [p_{i1}^{y_{i1}} (1 - p_{i1})^{1-y_{i1}} p_{i2}^{y_{i2}} (1 - p_{i2})^{1-y_{i2}}] \times \exp[-\frac{1}{2} \sum_{i=1}^n (\theta_i - \beta^T \eta_i(\gamma))^T \Sigma_e^{-1} (\theta_i - \beta^T \eta_i(\gamma))],$

where $(\dots)$ stands for the rest of the parameters and the data $\mathbf{y}$. The pdfs from (i) to (vi) are standard and it is easy to generate samples from them. The pdf in (viii) is log-concave so we can use adaptive rejection sampling of Gilks and Wild (1992) to generate samples from them. The pdf in (vii) is the most complicated and we need a Metropolis-Hastings algorithm of Chib and Greenberg (1995) to sample from it and update γ in blocks as suggested by Roberts and Sahu (1997).

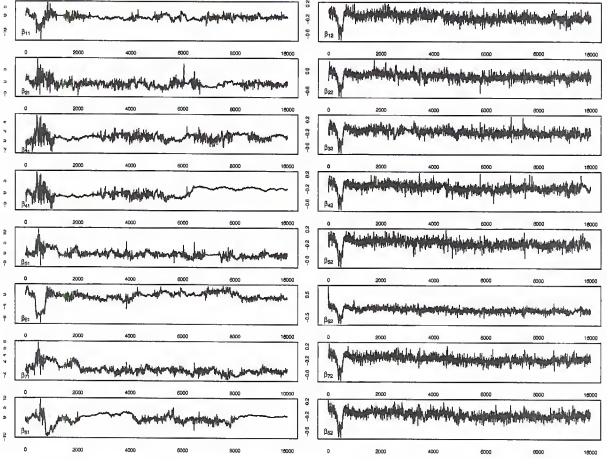


Figure 2-1: Trace plots of MCMC samples generated from the posterior distributions of β_1 in one chain for Example 1 and hyperparameter choice (i).

2.4 Applications to Prostate Cancer Study

2.4.1 Example 1: Application for Prostate Cancer with Clinical Covariates

The example studied here arises from a prostate cancer study, where 600 patients selected by simple random sample are used as the training cases and 234 separate patients are used as the test cases. We train our hierarchical Bayesian neural network with the training cases and use the trained neural network to predict the outcomes in the test cases. Here we know the actual truth for these 234 test cases; so after prediction we can check for the percentage of correct classification in the test set. Three clinical input factors or covariates are considered here. These are (i) unilateral vs bilateral tumor based on biopsy, (ii) gleason score measured in an ordinal scale of

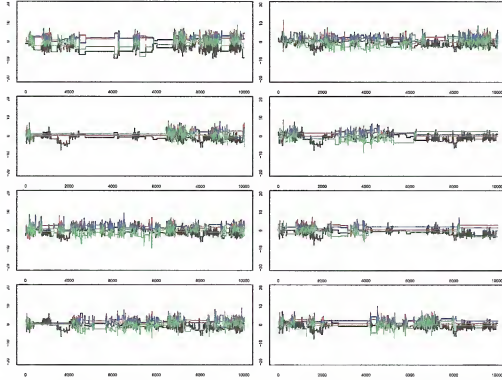


Figure 2-2: Trace plots of MCMC samples generated from the posterior distributions of γ_1 in one chain for Example 1 and hyperparameter choice (i).

2 to 10, and (iii) prostate specific antigen (PSA). The unilateral vs bilateral tumor is coded as binary 0 or 1 variable. We make a $\log(\text{PSA}+1)$ transformation on the raw PSA score. Along with this we add an intercept term so in our study there are $p = 4$ input nodes.

Here we fit a single layer feedforward neural network with fixed number of hidden nodes M . We tried several values of M and observed that the prediction error stabilizes at $M = 8$, so in the final model we use a neural network with 8 hidden nodes. Let $\mathbf{y}_u$ denote a future observation or an observation in the test set. The joint prediction probability of margin positivity and SV positivity is given by Equation 2-14.

$$P(y_{u1} = r, y_{u2} = s | \mathbf{y}) = E \left[\frac{\exp(r\theta_1 + s\theta_2)}{(1 + \exp(\theta_1))(1 + \exp(\theta_2))} | \mathbf{y} \right], \quad r, s = 0, 1. \quad (2-14)$$

The posterior expectation in Equation 2-14 is approximated by $\frac{1}{B} \sum_{i=1}^B \frac{\exp(r\theta_1^{(i)} + s\theta_2^{(i)})}{(1 + \exp(\theta_1^{(i)}))(1 + \exp(\theta_2^{(i)}))}$ where $\theta^{(i)}$ are generated from $N_2((\beta^{(i)})^T \eta(\gamma^{(i)}), \Sigma_e^{(i)})$, $\beta^{(i)}$, $\gamma^{(i)}$, $\Sigma_e^{(i)}$ are obtained from the i th MCMC sample, and B is the total number of such samples used. For this problem we used 5 independent chains each of length 50,000. To reduce the autocorrelation, every fifth sample is kept. Discard the first half as the burn in and use the last half of each chain and finally pool all the samples from the 5 different chains to produce the final estimate Gelman and Rubin (1992). In Figure 2-1., Figure 2-2., and Figure 2-3. We plot every fifth sample generated from the posteriors of the network parameters in one of the 5 chains. From the trace plots it is clearly seen that after an initial burn in of 20,000 iterations the MCMC stabilizes. For other chains also, MCMC behaves similarly and converges after the first 20,000 iterations on average. There are only 4 different values that $\mathbf{y}_u$ can take; $(0,0)^T$, $(1,0)^T$, $(0,1)^T$, $(1,1)^T$. Corresponding to a future observation $\mathbf{x}_u$ we compute the value of Equation 2-14 for all combinations of r and s and predict a $\mathbf{y}_u$ to take that particular combination for which the computed probability in Equation 2-14 is maximum.

For the choice of hyper-parameters, we set the values such that our priors are near-diffuse or vague but proper. In order to examine sensitivity in the choice of priors, we have considered several different choices of near-diffuse but proper priors. The misclassification rate remains stable with all such choices. Here we report the results for two such choices (i) $c_e = 5$, $C_e = 5I_2$; $c_{\beta_1} = c_{\beta_2} = 0.5$, $C_{\beta_1} = C_{\beta_2} = 15$; $c_{\gamma_1} = c_{\gamma_2} = 15$, $C_{\gamma_1} = C_{\gamma_2} = 16I_4$; $a_{\beta_1} = a_{\beta_2} = 0$, $A_{\beta_1} = A_{\beta_2} = 1000$; $a_{\gamma_1} = a_{\gamma_2} = 0$, $A_{\gamma_1} = A_{\gamma_2} = 100I_4$, and (ii) $c_e = 10$, $C_e = I_2$; $c_{\beta_1} = c_{\beta_2} = 0.05$, $C_{\beta_1} = C_{\beta_2} = 1$; $c_{\gamma_1} = c_{\gamma_2} = 10$, $C_{\gamma_1} = C_{\gamma_2} = I_4$; $a_{\beta_1} = a_{\beta_2} = 0$, $A_{\beta_1} = A_{\beta_2} = 1000$; $a_{\gamma_1} = a_{\gamma_2} = 0$, $A_{\gamma_1} = A_{\gamma_2} = 1000I_4$. Indeed, as we have made our priors near-diffuse or near-noninformative, i.e., priors with a very large variance, our method is not sensitive to the choice of prior parameters. If we assign priors with a tight support we might occasionally get a better prediction, but in that case the analysis will be highly

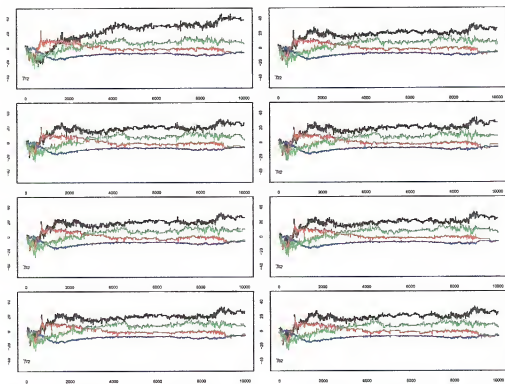


Figure 2-3: Trace plots of MCMC samples generated from the posterior distributions of γ_2 in one chain for Example 1 and hyperparameter choice (i).

Table 2-1: Percentage of correct prediction in the 234 test cases and 300 validation cases using our bivariate Hierarchical Bayesian Neural Network (HBNN) in Example 1.

	Hyperparameter Choice (i)			Hyperparameter Choice (ii)		
	$M = 4$	$M = 8$	$M = 12$	$M = 4$	$M = 8$	$M = 12$
Test Set	76.59	86.99	86.75	76.46	87.00	87.00
Validation Set	79.34	88.00	88.00	79.10	87.89	88.00

Table 2-2: Percentage of correct prediction in the 234 test cases and 300 validation cases using our bivariate Hierarchical Bayesian Neural Network (HBNN) as well as Ripley's Classical Neural Network (Ripley), Neal's Bayesian Neural Network (Neal), Bivariate Logistic Regression Models (BLM), and Univariate Hierarchical Bayesian Neural Network (UHBNN) in Example 1.

	HBNN	Ripley	Neal	BLM	UHBNN
Test Set	86.99	75.64	76.12	72.65	82.35
Validation Set	88.00	79.33	78.91	74.00	84.09

sensitive to the choice of the prior parameters. Also, with these type of near-diffuse proper priors, we are able to retain the objectivity of the method along with the propriety of the posterior.

Table 2-1 reports the percentage of correct joint prediction of MP and SV in the 234 test set samples with the hyperparameter values given in (i) and (ii) after we build our model on the 600 training samples. We say a prediction is correct if both SV positivity and MP are correctly predicted. For the choice of hidden nodes, we start with $M = 4$ hidden nodes, and after $M = 8$, we noticed that increasing the number of hidden nodes do not improve our prediction sufficiently relative to the increased computing time and complexity. So in the final model we keep $M = 8$. Table 2-1 lists the prediction accuracy of the model with 4, 8, and 12 hidden nodes, respectively.

Table 2-2 gives a comparison of our method with some other standard methods discussed in Section 2.2. Results based on our hierarchical Bayesian neural network (HBNN) are reported under the column HBNN. Percentage of correct joint prediction using other standard methods like Ripley's neural network, Neal's Bayesian neural

network and bivariate logistic models are also provided. Both in Neal's neural network and classical neural network, we used 8 hidden nodes similar to our Bayesian neural network. Ripley's R *nnet* routine of Venables and Ripley (1997) is used with the entropy fit option. As a neural network model may have several local minima, we have used 10 networks with different random starting points and averaged the prediction over all 10 networks as the final prediction. Results using Ripley's neural network are reported in Table 2-2 under the column name Ripley. For Neal's Bayesian neural network, we used his software with 3 inputs and 8 hidden units. We assigned a $N(0, 100^2)$ prior for the output bias and independent $N(0, \eta^{-1})$ prior for rest of the parameters. The precision parameter η has a gamma distribution with mean .05 and variance .01. More details about the prior specifications and the software can be obtained from Neal's web-site <http://www.cs.utoronto.ca/~radford/fbm.software.html>. Results using Neal's neural network is reported under the column name Neal. Neal's neural network is significantly different than what we have proposed here. Firstly Neal's model does not account for the error component ϵ_i . Secondly Neal's model fits the bivariate binary data without accounting for any dependence among them. So it is equivalent to fitting neural networks for each component of $\mathbf{y}_i$ separately. Instead our proposed method accounts for any dependence among the components of $\mathbf{y}_i$ and fits the network jointly. Also, unlike Neal, we assigned more flexible hierarchical priors on the network parameters. Bivariate logit model is fitted using the *VGAM* function in the R library and is reported under the name BLM. A detailed documentation along with the function can be obtained from the web-site of Thomas W. Yee, <http://www.stat.auckland.ac.nz/~yee>. We have also compared our method with the univariate hierarchical Bayesian neural network proposed by Ghosh et al. (2004). We apply their technique separately on MP and SV under similar near diffuse priors and

report them under the name UHBNN. Classify an observation as correctly predicted if MP and SV are both correctly predicted, otherwise the prediction is incorrect.

We brought in 300 new cases for model validation from a somewhat different population. All the methods discussed above were applied on this new set of data also. The outcome or prediction quality in the validation set is given in the second row of Table 2-1 and Table 2-2 under the name "Validation Set".

As observed from Table 2-2, for this specific problem, our hierarchical Bayesian neural network outperforms classical neural network, Neal's neural network as well as the bivariate logit model in predicting the margin positivity and seminal vesicle positivity jointly. The dependence among the components of $\mathbf{y}_i$ which we establish through our model helps to improve prediction over Neal's Bayesian neural network. Moreover, as discussed earlier, hierarchical Bayesian neural network helps us to get shrinkage estimators which borrow strength and produce better results than the classical estimators. Also, observe that by establishing the dependence among MP and SV we can increase our prediction accuracy over the one in Ghosh et al. (2004), which predicted MP and SV separately using univariate models.

2.4.2 Example 2: Application for Gene Expression Microarray Data

The recent advent of DNA microarray technique has made simultaneous monitoring of thousands of gene expressions possible. Staging and diagnosis based on gene expression profiles may provide more information than standard morphology, and thereby more accurately predict MP and SV positivity. This dataset comes from a study of gene expression in benign and malignant prostate tissue by Danasekaran et al. (2001). Gene expression levels were measured using a cDNA microarray containing 9,984 human genes. We have $n = 35$ observations. Initially we had many genes missing in a number of observations. We excluded all those genes from our analysis which are missing in more than ten percent of the prostate cancer samples and have low standard deviation across samples. After this initial gene filtering we

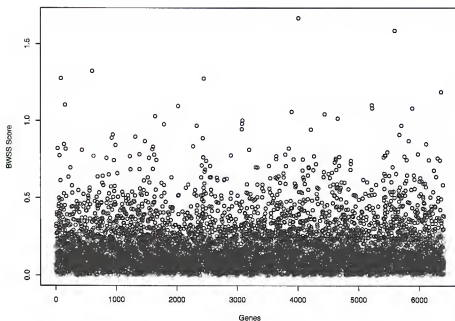


Figure 2-4: Marginal relevance of genes according to the between and within sum of squares criteria (BWSS) .

have $p = 6546$ genes. Among these 6546 genes, some are still missing for some of the observations. Impute the missing gene expression by the sample mean of that class. The gene expression data is summarized by an $n \times p$ matrix; so x_{ij} is the measurement of the expression level of the j th gene for the i th sample ($i = 1, \dots, n; j = 1, \dots, p$). Then take a log base 2 transformation on these expression levels x_{ij} , normalize and scale the log transformed measurements. In the response we have $\mathbf{Y}$ a bivariate binary vector which indicates the presence or absence of MP and SV in a patient.

Out of these 6546 genes, many genes do not contain information that is useful for determining the differences between the samples. These genes should not be used for prediction; also sometimes they may even contain noise that can lead to incorrect classification. Besides, the speed of training a neural network can be improved considerably by reducing the input information. It is impossible to train a neural network with so many input variables or covariates. Hence a simple criteria, proposed

by Dudoit et al. (2002) is used to rank the marginal relevance of each gene in class separation based on the ratio of between group and within group sums of squares, Figure 2.4. According to the values of $\mathbf{Y}$ i.e., $(0, 0)^T$, $(1, 0)^T$, $(0, 1)^T$, $(1, 1)^T$ we have 4 groups, 1, 2, 3, and 4 respectively. Let t_i denote the group of the i th patient. Hence t_i can take values 1, 2, 3, or 4. For a gene k , the ratio is

$$BW(k) = \frac{\sum_i \sum_h I(t_i = h)(\bar{x}_{hk} - \bar{x}_k)^2}{\sum_i \sum_h I(t_i = h)(\bar{x}_{ik} - \bar{x}_{hk})^2}$$

where $\bar{x}_k$ and $\bar{x}_{hk}$ denote the average expression level of gene k across all tumor samples and across samples belonging to class h only. After ranking the genes according to the criteria, select top 10 genes in our model and sequentially go on adding 10 sets of genes at a time following BW ordering criteria. We observe that the performance of our model is not greatly enhanced if more than 20 top genes are used. Rather sometimes due to some noise associated with some of the inactive genes, our model ends up giving more incorrect prediction. In Table 2-6 we give the crossvalidation error when different sets of genes are used. So in our final model top 20 genes are included and we have $p = 21$ input nodes including the intercept. As we have only 35 samples, instead of splitting the data into test and training sets, we check our model accuracy by leave one out cross-validation method. The prediction power can be increased by increasing the number of hidden nodes. In this example after 10 hidden nodes the cross-validation error remains the same, so finally we keep $M = 10$ in our model.

For choice of priors, we chose similar kind of near diffuse or vague but proper priors as we have done in the previous example. We explored several combinations of hyperparameters of the priors and found that near diffuseness of our prior makes our model very little sensitive to the choice of hyperparameters. The propriety of the posterior is guaranteed as we are using only proper priors. Here we report all our results in the light of these two choices of hyperparameters to validate our claim. (i)

Table 2-3: Leave one out cross-validation error of prediction using our bivariate Hierarchical Bayesian Neural Network (HBNN) in Example 2.

	$M = 5$	$M = 10$	$M = 20$
Hyperparameter Choice (i)	6	3	3
Hyperparameter Choice (ii)	5	3	4

Table 2-4: Leave one out cross-validation error using our bivariate Hierarchical Bayesian Neural Network (HBNN) as well as Ripley's Classical Neural Network (Ripley), Neal's Bayesian Neural Network (Neal), Bivariate Logistic Regression Models (BLM), and Univariate Hierarchical Bayesian Neural Network (UHBNN).

	HBNN	Ripley	Neal	BLM	UHBNN
Cross-validation error	3	7	6	8	6

$c_e = 5, C_e = 5I_2$; $c_{\beta_1} = c_{\beta_2} = 0.5, C_{\beta_1} = C_{\beta_2} = 15$; $c_{\gamma_1} = c_{\gamma_2} = 22, C_{\gamma_1} = C_{\gamma_2} = 25I_{21}$; $a_{\beta_1} = a_{\beta_2} = 0, A_{\beta_1} = A_{\beta_2} = 1000$; $a_{\gamma_1} = a_{\gamma_2} = 0, A_{\gamma_1} = A_{\gamma_2} = 100I_{21}$, and (ii) $c_e = 10, C_e = I_2$; $c_{\beta_1} = c_{\beta_2} = 0.05, C_{\beta_1} = C_{\beta_2} = 1$; $c_{\gamma_1} = c_{\gamma_2} = 30, C_{\gamma_1} = C_{\gamma_2} = I_{21}$; $a_{\beta_1} = a_{\beta_2} = 0, A_{\beta_1} = A_{\beta_2} = 1000$; $a_{\gamma_1} = a_{\gamma_2} = 0, A_{\gamma_1} = A_{\gamma_2} = 1000I_{21}$.

In Table 2-3, we report the leave one out crossvalidation error for the two choices of hyperparameters. We observe that the prediction error remains almost unchanged for the two different choices of hyperparameters, which indicates that our model is less sensitive to the choice of the prior parameters. The reason behind it is the near diffuseness of our chosen priors, which is explained in details in the previous example. In Table 2-4, we made a comparative study of the performance of our bivariate HBNN with the other standard methods as we did for the first example in Table 2-2. Our bivariate HBNN beats other methods like Neal's Bayesian neural network, bivariate logistic model, and Ripley's feedforward neural network by a considerably significant margin. Compared to the UHBNN of Ghosh et al. (2004), our bivariate method has much lower crossvalidation error of prediction, which indicates that introducing the dependence among the components helped us increase the prediction power of our model.

Table 2-5: Leave one out cross-validation error of prediction using our bivariate Hierarchical Bayesian Neural Network (HBNN) with clinical covariates in Example 2.

	$M = 5$	$M = 10$	$M = 20$
Hyperparameter Choice (i)	10	8	8
Hyperparameter Choice (ii)	9	9	8

Table 2-6: Leave one out cross-validation error of prediction using our bivariate HBNN in Example 2 with 10 hidden nodes.

	Number of genes included in the HBNN model				
	10	20	40	70	100
Hyperparameter Choice (i)	5	3	3	3	4
Hyperparameter Choice (ii)	5	4	4	4	4

To compare how the state of the art technology like gene expression microarray, improves the diagnosis and treatment for prostate cancer, we use clinical covariates like PSA and gleason score for the same 35 individuals for jointly predicting MP and SV. In Table 2-5, we report the leave one out crossvalidation error for the two choices of hyperparameters when clinical covariates are used instead of the microarrays. Comparing Table 2-3 and 2-5 we see that the gene expression profiles which provides more information than standard clinical covariates identify the MP and SV positivity more accurately. The crossvalidation errors of prediction using the clinical covariates are more than twice compared to the crossvalidation error using the microarray covariates.

2.4.3 Simulation Study

In order to simulate a realistic dataset for comparing, we used the prostate cancer data in Example 1 as a prototype. As realistic values of the network parameters Θ we used the posterior means from the original analysis of the data using the parameter choice (i). Then we followed the structure of our bivariate neural network model and performed one set of simulations to generate the responses $\mathbf{Y}$. We replicated each of the simulations 10 times, generating 10 different data sets. Then we analyze these training data sets using our bivariate HBNN and all other standard methods.

Table 2-7: Average percentage of correct prediction in the 234 test cases and 300 validation cases using our bivariate Hierarchical Bayesian Neural Network (HBNN) in Example 3.

	Hyperparameter Choice (i)			Hyperparameter Choice (ii)		
	$M = 4$	$M = 8$	$M = 12$	$M = 4$	$M = 8$	$M = 12$
Test Set	81.91	90.40	89.94	79.76	88.97	88.40
Validation Set	83.41	91.11	92.02	82.50	91.48	91.44

We calculate the average percentage of correct classification in the test set of size 234 and validation set of size 300 for all the models. Although we have generated data using hyperparameter choice (i) in Example 1, in the analysis stage we have considered both (i) and (ii) of prior parameters as given in the Example 1. This will help us understand specifically the effect of misspecification of priors in our analysis. In Table 2-7 we report the average percentage of correct classification. As in the previous examples, we notice here that our model is not very sensitive to the choice of prior parameters. It also indicates that prior misspecification does not greatly affect our prediction power as both choices of prior parameters yield very similar rates of correct classification. In Table 2-8 we report a comparison of our bivariate HBNN with other standard methods similar to Examples 1 and 2. From Table 2-8 we see that our bivariate HBNN is the best model in terms of average misclassification error in both the test set as well the validation set. On an average, the percentage of correct classification, based on our method is about ten percent higher than Ripley's neural network, Neal's Bayesian neural network, and BLM. When compared with the univariate version of HBNN, our bivariate extension achieves around five percent more accuracy in prediction.

2.5 Summary and Conclusion

In this chapter, we have introduced a hierarchical Bayesian neural network model for the analysis of bivariate binary data. Rather than a deterministic search for minimization of some error measure, it uses a stochastic search method motivated by a Bayesian model, and uses the posterior predictive distribution to predict a future

Table 2–8: Percentage of correct prediction in the 234 test cases and 300 validation cases using our bivariate Hierarchical Bayesian Neural Network (HBNN) as well as Ripley’s Classical Neural Network (Ripley), Neal’s Bayesian Neural Network (Neal), Bivariate Logistic Regression Models (BLM), and Univariate Hierarchical Bayesian Neural Network (UHBNN) in Example 3.

	HBNN	Ripley	Neal	BLM	UHBNN
Test Set	90.40	78.89	80.42	77.93	85.47
Validation Set	91.11	82.61	79.94	76.20	86.42

observation. Unlike Neal’s method or classical neural network approach our method establishes a dependence among the components of the output variable, builds a much stronger learning algorithm, and finally has better prediction power. This is revealed in the two examples and also in the simulation study given in 2.4.1 – 2.4.3. Bayesian learning integrates over the posterior distribution for the network parameters, rather than picking a single “optimal” set of parameters. This avoids the problem of overfitting. Using a hierarchical prior, we can automatically determine how relevant each input or covariate is in predicting the MP and SV. If a covariate is irrelevant in predicting the margin positivity and seminal vesicle positivity, the hyperparameters corresponding to its weight will tend to be small, thus forcing the weight to be near zero. Moreover we may also point out that the Bayesian approach addresses the problem in a pure probabilistic framework, rather than a simple function optimization approach. Thus, we can obtain the predictive distributions for the joint occurrence of margin and SV positivity rather than simple point estimates.

We have used both clinical and gene expression microarray covariates in our neural network model, and have found the outcomes to be superior to other competing methods. It is particularly noticed that in a gene expression microarray study where the sample size is very small, our bivariate HBNN is the most effective and accurate compared to other standard methods. In the gene expression microarray example, our method outperforms other discussed methods by a wide margin, validating our claim. Selecting differentially genes is very important for the performance of our bivariate

HBNN predictor and utmost care must be taken to identify the active genes from the nonactive genes. We have also found that our model is not sensitive to the choice of (near diffuse) hyperparameters. For example, in the simulation study, the conclusion is nearly the same even with a misspecified prior. This shows the objectivity of our method and its wide range of applications.

If cancer is found in the prostate, physicians try to determine the stage, or the extent, of the disease. This method can help the doctors to identify the stage of prostate cancer much more accurately, by jointly predicting certain features like margin positivity and seminal vesicle positivity, which are strong indicators of non-organ confined cancer. This can avoid radical interventions in potentially incurable patients. Compared to other methods like bivariate logistic regression, classical neural network, and Neal's Bayesian neural network, our method provides a much higher rate of accurate predictions in all the prostate cancer datasets discussed earlier. Along with this, ours is a first attempt to predict the margin positivity and seminal vesicle positivity simultaneously in a patient suffering from prostate cancer.

The models considered so far are fully parametric. It is possible to enrich the proposed class of models by adopting a semiparametric hierarchical Bayesian approach. Moreover, as a general methodology (but not for this particular problem), we can consider a variable number of nodes Rios and Müller (1998), and assign a Poisson or negative binomial prior to the number of nodes. A reversible jump MCMC algorithm can be devised to select the number of nodes. Apart from this, a stochastic search variable selection method can also be integrated with our bivariate HBNN model to select differentially expressed gene and predict the margin and SV positivity simultaneously.

CHAPTER 3 BAYESIAN NONLINEAR REGRESSION FOR LARGE p SMALL n PROBLEMS

“Prediction is very difficult, especially if it’s about the future.” —Niels Bohr

Regression techniques are amongst some of the most widely used methods in applied statistics. Given a response variable Y , and a set of covariates $\mathbf{X} = (X_1, X_2, \dots, X_p)^T$, one is often interested in predicting further responses for new values of the covariates. The problem becomes complicated when the sample size n is substantially smaller than the number of covariates p . This is known as the large p small n problem or high dimension low sample size problem. The usual way to handle the problem is to reduce the number of covariates by using variable selection (Brown et al., 1999) or projecting them to lower dimension using principal component or other related methods (West, 2003).

Most of the existing methods for variable selection or projections are based on linear relationship between the response and the covariates which may not be very realistic. Recent, advances in computing power have allowed statisticians to consider richer classes of nonlinear regression models. Due to the large p small n problem, these methods are usually inapplicable in these situations. We use instead the reproducing kernel Hilbert space (RKHS) methodology for these problems. We are particularly interested in the analysis of multivariate data with correlated responses and extend the univariate nonlinear prediction models in this setup.

One important motivating application arises from the near infrared (NIR) spectroscopy. An experiment involving spectral measurements typically produces more covariates (wavelengths/channels) than calibration measurements (samples). Traditional stepwise regression methods fail miserably in these circumstances because

of the lack of necessary samples. Chemometricians often use principal components regression (PCR), partial least squares (PLS) regression, multiple linear regression (discussed in Chapter 1) to overcome these limitations. Both principal components regression and partial least squares regression, produce factor scores as linear combinations of the original predictor variables, so that there is no correlation between the factor score variables used in the predictive regression model. But they differ in the methods used in extracting the factor scores. In short, PCR produces the weight matrix reflecting the covariance structure between the predictor variables, while PLS regression produces the weight matrix reflecting the covariance structure between the predictor and response variables. We know from chemistry and optics that reflectance or absorbencies are linearly related to the concentrations, (Beer-Lambert Law). There is a deviation from this law for reasons such as optical scattering, autofluorescence. The resultant nonlinearities impair the accuracy of prediction based on linear relationships. The problem becomes more complicated when there are multiple constituents under investigation, as we then need to develop multivariate non linear regression models for prediction.

One of the models which we will consider for this purpose is support vector machine (SVM). The classical support vector machine (SVM) algorithm, despite its success in regression and classification, suffers from a serious disadvantage: it cannot provide any probabilistic outputs. Law and Kwok (2001) introduced a Bayesian formulation of SVM for regression, but they did not carry out a full Bayesian analysis and used instead certain approximations for the posterior. A similar remark applies to Sollich (2001) who considered SVM in the classification context. More recently, Mallick et al. (2005) considered a Markov chain Monte Carlo (MCMC) based full Bayesian analysis for classification where the number of covariates was far greater than the sample size.

As an alternative to SVM, Tipping (2000, 2001) and Bishop and Tipping (2000) introduced relevance vector machines (RVM's). RVM's are suitable for both regression and classification, and are amenable to probabilistic interpretation. However, these authors did not perform a full Bayesian analysis. They obtained type II maximum likelihood (Good, 1965) estimates of the prior parameters, which do not provide predictive distribution of the future observations. Also, their procedure cannot provide measures of precision associated with the estimates.

The present chapter addresses a full Bayesian analysis of regression problems when p , the number of covariates far outnumber n , the sample size. The SVM approach is based on reproducing kernel Hilbert spaces (RKHS) as well as Vapnik's (1995) ϵ -insensitive loss function. We also consider the RVM-based analysis in this context, once again by using RKHS. Ours is a full hierarchical Bayesian model. Instead of relying on the type II maximum likelihood estimates of the prior parameters, we assign distributions to these parameters. In this way, we can capture the uncertainty in estimating these parameters, and consequently get more reliable measures of precision associated with the Bayes estimates. A key feature of our method is to treat the kernel parameter in the model as unknown and infer about it with all other parameters. We obtain a more accurate prediction by the mixing of several RVM (or SVM) models through the kernel parameter. Due to analytical intractability of posteriors, we use the MCMC numerical integration technique for implementation of the Bayes procedure.

The methods are applied for prediction of glucose concentration in blood based on fluorescence-based optics. The example is first considered by Spiegelman et al. (2002). Their objective was to linearize nonlinear spectra for calibration, showing in the process that their proposed calibration was comparable in terms of prediction to partial least squares (PLS) and principal component regression (PCR). Our objective

also is to predict the future responses of glucose concentration. It turns out that our method is superior to PLS, PCR as well as the classical support vector regression.

Next we extend both the RVM and SVM methodology to the case when the response is multivariate. This requires extension of the univariate models that have been proposed thus far, and broadens the scope of our application. The methods are illustrated with a real data set, discussed by Brown et al. (2001). The available measurements are the near infrared reflectance spectrum of biscuit dough pieces at 700 equally spaced wavelengths. The aim is to predict the composition i.e., the fat, sugar, flour, and water content of the dough pieces using the spectral variables. We compare our procedure with some of the classical methods like stepwise multiple linear regression (MLR), PLS, PCR, and classical support vector regression as well as Bayesian wavelet regression (Brown et al., 2001) based on mean square errors of prediction (MSEP). Univariate SVM and RVM are also applied separately on each constituent. It turns out that, our multivariate Bayesian RVM performs better than all other methods, for all the components except fat, where stepwise MLR has the best performance.

Apart from introducing hierarchical Bayes RVM and SVM models, we have also suggested an empirical Bayes analysis for RVM and SVM models in both univariate and multivariate setup. In univariate setup, our empirical Bayes RVM is similar to the RVM model proposed by Tipping (2001). Unlike Tipping we are not keeping the kernel parameter fixed in our model. We are rather estimating the kernel parameter from the data by maximizing the marginal posterior. A similar approach is taken for empirical Bayes SVM. The empirical Bayes analysis is also extended in the multivariate framework, and the results are tabulated in Tables 4-1 to 4-4.

Section 3.2 of this chapter introduces the RKHS-based regression method. The hierarchical Bayes RVM and SVM models for regression is discussed in details in Sections 3.3 and 3.4 respectively. The multivariate Bayesian RVM and SVM are

introduced in Sections 3.5 and 3.6. Section 3.7 discusses prediction of a future observation. Section 3.8 illustrates our methodology developed in Sections 3.3-3.7 with two real life datasets and a simulated dataset. In Section 3.9 we introduce empirical Bayes SVM and RVM. Finally, some concluding remarks are made in Section 3.10.

3.1 Regression Method Based on Reproducing Kernel Hilbert Space

In Section 1.4 we have introduced a general way of casting the regression problem in a regularization framework. For a regression problem, we have a training set $\{y_i, \mathbf{x}_i\}$, $i = 1, \dots, n$, where y_i is the response variable and $\mathbf{x}_i$ is the vector of covariate of size p corresponding to y_i . Let us consider that the response y and the covariate $\mathbf{x}$ are related by an unknown function $f(\mathbf{x})$ as in Equation 3-1.

$$\mathbf{y} = f(\mathbf{x}) + \epsilon. \quad (3-1)$$

Where ϵ is the error associated with our model. We build our regression model on the training data and then find the appropriate regression function $f(\mathbf{x})$ to predict the responses y in the test set based on the covariates $\mathbf{x}$. This can be viewed as a regularization problem of the form as before in Section 1.4 in Chapter 1. Like before we will consider that the unknown function $f(\mathbf{x})$ belongs to the reproducing Kernel Hilbert space (RKHS) of functions, denoted by $\mathcal{H}_K$. A formal definition of RKHS is given in Aronszajn (1950), Parzen (1970), and Wahba (1990).

Hence for an $h \in \mathcal{H}_K$, if $f(\mathbf{x}) = \beta_0 + h(\mathbf{x})$, by the Wahba representation in Equation 1-21 as

$$f_\lambda(\mathbf{x}) = \beta_0 + \sum_{i=1}^n \beta_i K(\mathbf{x}, \mathbf{x}_i) \quad (3-2)$$

where $K(\cdot, \cdot)$ is the kernel function and λ is the smoothing parameter. When the number of covariates p is much larger than n by the above Wahba representation of f in above form, we effectively reduce the dimension of covariates from p to $n + 1$. We estimate the β and λ by minimizing the corresponding penalized loss function in

Equation 3-3.

$$\min_{f \in \mathcal{H}_K} \left[\sum_{i=1}^n L(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}_K}^2 \right]. \quad (3-3)$$

Similarly, for multivariate regression when $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_q(\mathbf{x}))$ is a q -tuple function we can have similar results based on RKHS as the univariate case. Here we consider $f(\mathbf{x}) \in \prod_{j=1}^q (\{1\} + \mathcal{H}_{K_j})$, the product space of q reproducing kernel Hilbert spaces $\mathcal{H}_{K_j}$ for $j = 1, \dots, q$. In other words, each component can be expressed as $\beta_j + h_j(\mathbf{x})$ with $h_j \in \mathcal{H}_{K_j}$. Unless there is a compelling reason to believe that $\mathcal{H}_{K_j}$ should be different for $j = 1, \dots, q$, we will assume that they are the same RKHS denoted by $\mathcal{H}_K$. All the results stated before hold in this framework as well.

Regarding the loss $L(y, f(\mathbf{x}))$ as the negative of the log-likelihood, our problem is equivalent to maximization of the penalized log-likelihood in Equation 3-4.

$$-\sum_{i=1}^n L(y_i, f(\mathbf{x}_i)) - \lambda \|h\|_{\mathcal{H}_K}^2. \quad (3-4)$$

This duality between “loss” and “likelihood”, particularly viewing the loss as the negative of the log-likelihood, is referred in the Bayesian literature as the “logarithmic scoring rule” (Bernardo, 1979, p. 688).

3.2 Hierarchical Bayes Relevance Vector Machine

The support vector machine (SVM) is a highly sophisticated technique for regression and classification. It is well known that SVM can be derived as the minimizer of the function $\sum_{i=1}^n L(y_i, f(\mathbf{x}_i)) + \lambda \|h\|_{\mathcal{H}_K}^2$ in RKHS (Wahba, 1999). Despite its widespread success, the classical SVM suffers from some important limitations, notably the absence of probabilistic output i.e it makes point predictions rather than generating predictive distributions. Recently Tipping (2000) has formulated the relevance vector machine (RVM), a probabilistic model whose functional form is equivalent to the SVM with a least square loss function (also known as LS-SVM). Law and Kwok (2001) considered a Bayesian SVM model with Vapnik's ϵ -insensitive robust loss function, a detailed discussion is provided in the next section.

The original treatment of RVM relied on the use of type II maximum likelihood (Good, 1965) estimates of the hyper-parameters. In this section we formulate RVM and in the next section we formulate SVM in a complete hierarchical Bayesian paradigm based on the RKHS representation, as discussed in the previous section.

If we assume that f is generated from RKHS with the kernel function $K(.,.)$, using the representation theorem (Equation 1-21) we can express f as in Equation 3-5.

$$f(\mathbf{x}_i) = \beta_0 + \sum_{j=1}^n \beta_j K(\mathbf{x}_i, \mathbf{x}_j | \theta) \quad (3-5)$$

where K is a positive definite function of the covariates $\mathbf{x}$ involving some unknown parameter θ . Hence,

$$y_i = f(\mathbf{x}_i) + \eta_i = \mathbf{K}_i^T \boldsymbol{\beta} + \eta_i \quad (3-6)$$

where $\eta_i \stackrel{iid}{\sim} N(0, \sigma^2)$, $\boldsymbol{\beta} = (\beta_0, \dots, \beta_n)^T$, and $\mathbf{K}_i = (1, K(\mathbf{x}_i, \mathbf{x}_1 | \theta), \dots, K(\mathbf{x}_i, \mathbf{x}_n | \theta))^T$, $i = 1, \dots, n$. We assign hierarchical priors to the unknown parameters $\boldsymbol{\beta}$, θ , and σ^2 . A conjugate prior for this model is given in Equation 3-7.

$$\boldsymbol{\beta} | \sigma^2 \sim N_{n+1}(0, \sigma^2 \mathbf{D}^{-1}) \quad (3-7)$$

where $\mathbf{D} = \text{diag}(\lambda_0, \dots, \lambda_n)$ is a $(n+1) \times (n+1)$ diagonal matrix and $\boldsymbol{\lambda} = (\lambda_0, \dots, \lambda_n)^T$. λ_0 is fixed at a small value but other λ 's are kept unknown. We also assume that

$$\sigma^2 \sim \text{IG}(\gamma_1, \gamma_2). \quad (3-8)$$

A Gamma(α, ξ) distribution for a random variable U has density function proportional to $\exp(-\xi u) u^{\alpha-1}$, while the reciprocal of U will then said to have a inverse gamma distribution denoted by IG(α, ξ). We also assume that,

$$\theta \sim U(a_L, a_U), \quad \lambda_i \stackrel{iid}{\sim} \text{Gamma}(c, d), \quad i = 1, \dots, n \quad (3-9)$$

where $U(a_L, a_U)$ is the uniform distribution over (a_L, a_U) .

The matrix with (i, j) th element $K_{ij} = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j | \theta)$ is known as the kernel matrix. The usual choices are (i) the Gaussian kernel $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j | \theta) = \exp\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\theta}\}$ and (ii) the polynomial kernel, $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j | \theta) = (\mathbf{x}_i^T \mathbf{x}_j + 1)^\theta$. Both kernels contain a single parameter θ .

The joint posterior distribution is thus given by

$$\begin{aligned} \pi(\beta, \lambda, \sigma^2, \theta | \mathbf{y}) &\propto \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left\{-\sum_{i=1}^n \frac{(y_i - \mathbf{K}_i^T \beta)^2}{2\sigma^2}\right\} \\ &\times \frac{1}{(2\pi)^{(n+1)/2} |\sigma^2 \mathbf{D}^{-1}|^{1/2}} \exp\left(-\frac{1}{2\sigma^2} \beta^T \mathbf{D} \beta\right) \\ &\times \exp(-\gamma_2/\sigma^2) (\sigma^2)^{-\gamma_1-1} \\ &\times \prod_{i=1}^n \exp(-d\lambda_i) \lambda_i^{c-1}. \end{aligned} \quad (3-10)$$

This distribution is complex, and implementation of the Bayesian procedure requires MCMC sampling techniques, and in particular, the Gibbs sampling (Gelfand and Smith, 1990) and Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953, Chib and Greenberg, 1995). The Gibbs sampler generates posterior samples using the conditional densities of the model parameters which we describe below.

Conditional Distributions and Posterior Sampling of the Parameters.

The prior distributions in Equation 3-7 and Equation 3-8 are conjugate distributions for β and σ^2 . The posterior density conditional on θ , λ given in Equation 3-11 is Normal-Inverse-Gamma.

$$p(\beta, \sigma^2 | \theta, \lambda) = N_{n+1}(\beta | \tilde{\mathbf{m}}, \sigma^2 \tilde{\mathbf{V}}) \text{IG}(\sigma^2 | \tilde{\gamma}_1, \tilde{\gamma}_2) \quad (3-11)$$

where $\tilde{\mathbf{m}} = (\mathbf{K}_0^T \mathbf{K}_0 + \mathbf{D})^{-1} (\mathbf{K}_0^T \mathbf{y})$, $\tilde{\mathbf{V}} = (\mathbf{K}_0^T \mathbf{K}_0 + \mathbf{D})^{-1}$, $\tilde{\gamma}_1 = \gamma_1 + n/2$, and $\tilde{\gamma}_2 = \gamma_2 + (\mathbf{y}^T \mathbf{y} - \tilde{\mathbf{m}}^T \tilde{\mathbf{V}} \tilde{\mathbf{m}})$. Here $\mathbf{K}_0^T = (\mathbf{K}_1, \dots, \mathbf{K}_n)$.

The conditional distribution for λ_i given β_i and σ^2 is given by

$$\lambda_i | \beta_i, \sigma^2 \stackrel{\text{ind}}{\sim} \text{Gamma}\left(c + \frac{1}{2}, d + \frac{\beta_i^2}{2\sigma^2}\right), \quad i = 1, \dots, n. \quad (3-12)$$

We use the conditional distributions for constructing a Gibbs sampler through following steps:

- Step 1. Update β and σ^2 by sampling from their conditional distributions (Equation 3-11).
- Step 2. Update λ by sampling from the conditional distribution (Equation 3-12) of λ_i in turn.
- Step 3. Update of K is equivalent to the update of θ and we need a Metropolis-Hastings algorithm to sample from the marginal distribution of θ . We put an uniform prior on θ so $p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta)$, the marginal likelihood of θ . Let θ^* denote the proposed change. Accept this change with acceptance probability

$$\alpha = \min \left\{ 1, \frac{p(\mathbf{y}|\theta^*)}{p(\mathbf{y}|\theta)} \right\}. \quad (3-13)$$

The ratio of the marginal likelihoods in Equation 3-14 is given by

$$\frac{p(\mathbf{y}|\theta^*)}{p(\mathbf{y}|\theta)} = \frac{|\tilde{\mathbf{V}}^*|^{1/2}}{|\tilde{\mathbf{V}}|^{1/2}} \left(\frac{\hat{\gamma}_2}{\hat{\gamma}_2^*} \right)^{\tilde{\gamma}_1}. \quad (3-14)$$

Using the marginal posteriors rather the full conditional posterior improves the mixing by a large extent. Also by integrating out other parameters, a much faster convergence of the chain is obtained when compared to using the full conditional. Stability of the chain is also increased without compromising the mixing. This method of using the marginal distribution of θ , the kernel parameter, for sampling, is also successfully used by Mallick et al. (2005).

In the relevance vector machine discussed above, we effectively try to minimize the squared error loss function and use separate smoothing parameters λ_i for different β_i . The smoothing parameters determine the trade-off between training accuracy and model complexity. We can also keep all λ_i 's same by fixing $\lambda_i = \lambda$, for all $i = 1, \dots, n$. This resulting approach will be referred to as the Bayesian relevance vector machine (BRVM). The multiple smoothing parameter is also used by Tipping (2000). By

having multiple smoothing parameters over a single one we effectively control for each of the regression coefficients separately, thereby introducing sparseness in the model.

3.3 Hierarchical Bayes Support Vector Machine

In this section, we show how SVM based on RKHS can be used in a complete Bayesian setup using Vapnik's ϵ -insensitive loss function.

The ϵ -insensitive loss function introduced by Vapnik (1995) is as shown in Equation 3-15.

$$L(y, f(\mathbf{x})) = |y - f(\mathbf{x})|_\epsilon = \begin{cases} 0 & \text{if } |y - f(\mathbf{x})| \leq \epsilon \\ |y - f(\mathbf{x})| - \epsilon & \text{otherwise.} \end{cases} \quad (3-15)$$

This loss function (Figure 3-1) ignores errors of size less than ϵ but penalizes in a linear fashion when the function deviates more than ϵ amount. This makes the fitting less sensitive to the outliers. It is interesting to note that like other loss functions or error measures in robust regression (Huber, 1964), Vapnik's ϵ -insensitive loss function also has linear tails beyond ϵ . But in addition it flattens the contributions of those cases with small residuals.

To construct a hierarchical model for regression using Vapnik's loss function we introduce n latent variables $z_1, \dots, z_n$. Such that y_i 's are conditionally independent given z_i . The introduction of the latent variables makes the calculations particularly simple. The conditional likelihood of $y_i|z_i$, corresponding to Vapnik's loss in Equation 3-15 is given by Equation 3-16.

$$p(y_i|z_i) \propto \exp\{-\rho|y_i - z_i|_\epsilon\}, \quad i = 1, \dots, n. \quad (3-16)$$

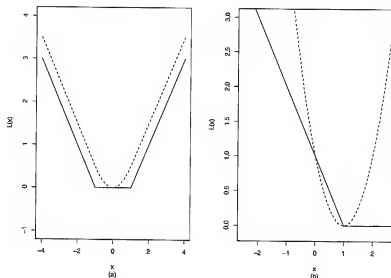


Figure 3-1: The figure in the left hand side represents Vapnik's ϵ -insensitive ($\epsilon = 1$) loss function. The figure in the right hand side represents the corresponding likelihood.

It can be shown that this likelihood (Figure 3-1) can be written as a mixture of a truncated Laplace distribution and a uniform distribution as follows:

$$\begin{aligned}
 p(y_i|z_i) &= \frac{\rho}{2(1+\epsilon\rho)} \exp\{-\rho|y_i - z_i|\epsilon\} \\
 &= \frac{\rho}{2(1+\epsilon\rho)} [I(|y_i - z_i| > \epsilon) \exp\{-\rho(|y_i - z_i| - \epsilon)\} + I(|y_i - z_i| \leq \epsilon)] \\
 &= p_1(\text{Truncated Laplace}(z_i, \rho)) + p_2(\text{Uniform}(z_i - \epsilon, z_i + \epsilon)) \quad (3-17)
 \end{aligned}$$

where $p_1 = \frac{1}{1+\epsilon\rho}$, $p_2 = \frac{\epsilon\rho}{1+\epsilon\rho}$.

A $\text{Laplace}(\theta, \rho)$ distribution for a random variable U has pdf proportional to $\exp(-\rho|u - \theta|)$.

We connect z_i with $f(\mathbf{x}_i)$ by $z_i = f(\mathbf{x}_i) + \eta_i$, where η_i are the residual random effects that account for any unexplained source of variation not included in the model. We assume that f is generated from RKHS. By the representation theorem (Equation 1-21), we can express f as in Equation 3-5. So the z_i 's are modeled as

$$z_i = \mathbf{K}_i^T \boldsymbol{\beta} + \eta_i \quad (3-18)$$

where $\eta_i \stackrel{iid}{\sim} N(0, \sigma^2)$ and β and K_i defined as in Section 3.3. We assign hierarchical priors to the unknown parameters $\beta, \theta, \rho, \sigma^2, z_i, \lambda_i$ as follows.

$$z_i | \beta, \sigma^2, \theta, \lambda \stackrel{ind}{\sim} N(K_i^T \beta, \sigma^2) \quad (3-19)$$

$$\beta | \sigma^2, \lambda \sim N(0, \sigma^2 D^{-1}); D = \text{diag}(\lambda_0, \lambda_1, \dots, \lambda_n) \quad (3-20)$$

$$\sigma^2 \sim \text{IG}(\gamma_1, \gamma_2) \quad (3-21)$$

$$\theta \sim U(a_L, a_U), \lambda_i \stackrel{iid}{\sim} \text{Gamma}(c, d), \rho \sim U(r_L, r_U). \quad (3-22)$$

Asymptotically if ϵ goes to infinity, we get the uniform likelihood, but if it goes to 0 we get the usual Laplace likelihood when ρ is fixed. So instead of keeping ϵ fixed, as in classical SVM we assigned prior to ϵ . However we observed that the performance of SVM decayed rapidly for priors spreading outside the range $(0, 1)$. More detailed justifications are provided in Law and Kwok (2001). Hence we have considered

$$\epsilon \sim \text{Beta}(k_1, k_2). \quad (3-23)$$

This Bayesian SVM model bears some analogy to the classical SVM as the exponent of the Gaussian prior for β is equivalent to the quadratic penalty function, but with multiple smoothing parameters. The posterior is similar to the posterior for the RVM model, except now the Gaussian likelihood in RVM is changed to Vapnik's ϵ -insensitive loss based likelihood (Equation 3-16). In Section 3.7.2 we will see that the likelihood corresponding to the robust ϵ -insensitive loss help us to handle situations involving outliers in the data.

The joint posterior is given by Equation 3-24.

$$\begin{aligned}
\pi(\beta, \lambda, \mathbf{z}, \sigma^2, \theta, \rho, \epsilon | \mathbf{y}) &\propto \frac{\rho^n}{2^n(1+\epsilon\rho)^n} \exp\left(-\rho \sum_{i=1}^n |y_i - z_i|_\epsilon\right) \\
&\times \frac{1}{(2\pi)^{n/2}(\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (z_i - \mathbf{K}_i^T \beta)^2\right) \\
&\times \frac{1}{(2\pi)^{(n+1)/2} |\sigma^2 \mathbf{D}^{-1}|^{1/2}} \exp\left(-\frac{1}{2\sigma^2} \beta^T \mathbf{D} \beta\right) \\
&\times \exp\left(-\frac{\gamma_2}{\sigma^2}\right) (\sigma^2)^{-\gamma_1-1} \\
&\times \prod_{i=1}^n \exp(-d\lambda_i) \lambda_i^{c-1} \\
&\times \epsilon^{k_1-1} (1-\epsilon)^{k_2-1}.
\end{aligned} \tag{3-24}$$

The posterior distribution is very complex and implementation of Bayesian methods is done once again using MCMC. The conditional posterior distribution of λ_i is exactly the same as in the RVM case given by Equation 3-12 in Section 3.2.1. The conditional distribution for β and σ^2 is also similar to that in the previous section, except that now $\mathbf{y}$ is replaced by the latent variable $\mathbf{z}$ in $\tilde{\mathbf{m}}$ and in $\tilde{\gamma}_2$ in Equation 3-11.

The distribution of z_i , conditional on $\mathbf{y}, \mathbf{K}_i, \beta, \theta, \rho, \lambda, \epsilon$, and $\mathbf{z}_{-i}$ ($\mathbf{z}_{-i}$ indicates the $\mathbf{z}$ vector with the i th element removed) does not have an explicit form. We thus resort to Metropolis Hastings (MH) algorithm with a proposal density $T(z_i^* | z_i)$ that generates moves from the current state z_i to a new state z_i^* . It is convenient to take the proposal distribution to be Gaussian with mean equal to $\mathbf{K}_i^T \beta$ and variance σ^2 (Chib and Greenberg, 1995). The proposed updates are then accepted with probabilities

$$\begin{aligned}
\delta_i &= \min \left\{ 1, \frac{p(z_i^* | \mathbf{z}_{-i}, \mathbf{K}, \sigma^2, \beta, \theta, \rho, \epsilon, \mathbf{y}) T(z_i | z_i^*)}{p(z_i | \mathbf{z}_{-i}, \mathbf{K}, \sigma^2, \beta, \theta, \rho, \epsilon, \mathbf{y}) T(z_i^* | z_i)} \right\} \\
&= \min \left\{ 1, \frac{\exp(-\rho |y_i - z_i^*|_\epsilon)}{\exp(-\rho |y_i - z_i|_\epsilon)} \right\}.
\end{aligned} \tag{3-25}$$

The update of $\mathbf{K}$ or θ is done similarly as before using the MH algorithm. Instead of sampling from the full marginal of θ we sample from $p(\theta | \mathbf{y}, \mathbf{z}, \rho, \epsilon)$, the marginal posterior of θ integrating out λ, β, σ^2 . If θ^* denotes the proposed change from current

θ , we accept this θ^* with probability

$$\alpha = \min \left\{ 1, \frac{p(\theta^* | \mathbf{z}, \mathbf{y}, \rho, \epsilon)}{p(\theta | \mathbf{z}, \mathbf{y}, \rho, \epsilon)} \right\}. \quad (3-26)$$

The ratios of the marginal posteriors are same as in Equation 3-14 except that $\mathbf{y}$ is now replaced by the latent variable $\mathbf{z}$ in the expression.

The conditional distribution of ρ depending only on the latent variable $\mathbf{z}$, ϵ and the data $\mathbf{y}$ is given by Equation 3-27.

$$p(\rho | \mathbf{y}, \mathbf{z}, \epsilon) \propto \frac{\rho^n}{(1 + \epsilon\rho)^n} \exp \left(-\rho \sum_{i=1}^n |y_i - z_i|_\epsilon \right). \quad (3-27)$$

The conditional distribution of ϵ conditional on ρ , $\mathbf{y}$, and $\mathbf{z}$ is given by Equation 3-28.

$$p(\epsilon | \mathbf{y}, \mathbf{z}, \rho) \propto \frac{\rho^n}{(1 + \epsilon\rho)^n} \exp \left[-\rho\epsilon \sum_{i=1}^n I(|y_i - z_i| > \epsilon) \right] \times \epsilon^{k_1-1} (1 - \epsilon)^{k_2-1}. \quad (3-28)$$

We use the MH algorithm to generate ρ and ϵ respectively from Equation 3-27 and Equation 3-28 with the acceptance probabilities $\left\{ 1, \frac{p(\rho^* | \mathbf{y}, \mathbf{z}, \epsilon)}{p(\rho | \mathbf{y}, \mathbf{z}, \epsilon)} \right\}$ and $\left\{ 1, \frac{p(\epsilon^* | \mathbf{y}, \mathbf{z}, \rho)}{p(\epsilon | \mathbf{y}, \mathbf{z}, \rho)} \right\}$ respectively.

Using the above conditional distributions, we can construct a Gibbs sampler by following the same steps 1 to 3 as in RVM. The distributions of some of the conditionals are changed as we replace $\mathbf{y}$ by the latent variable $\mathbf{z}$. Three extra steps needed to generate the latent variables $\mathbf{z}$, ρ and ϵ are added as follows:

Step 4. We update each z_i in turn conditional on the rest using the Metropolis step discussed in Equation 3-25.

Step 5. Update ρ using a Metropolis step involving Equation 3-27.

Step 6. Update ϵ using a Metropolis step involving Equation 3-28.

The resulting support vector machine discussed above is based on the likelihood corresponding to the Vapnik's ϵ -insensitive loss function (Equation 3-15) with multiple smoothing parameters λ_i . This resulting approach will be referred as the Bayesian support vector machine (BSVM). As in the case of BRVM, here also we

can also simplify our model by fixing same smoothing parameter, $\lambda_i = \lambda$ for all components of β .

3.4 Hierarchical Bayes RVM for Multivariate Regression

In this section, we extend the formulation of RVM for regression as in Section 3.3 to when the responses are multivariate.

Let us define the data set $\mathcal{D} = ((\mathbf{y}_1, \mathbf{x}_1), \dots, (\mathbf{y}_n, \mathbf{x}_n))$, consisting of multivariate responses $\mathbf{y}_i = (y_{i1}, \dots, y_{iq})^T$ and explanatory variable $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$, $i = 1, \dots, n$. We introduce n latent variables $\mathbf{z}_1, \dots, \mathbf{z}_n$, where $\mathbf{z}_i = (z_{i1}, \dots, z_{iq})^T$ and assume that conditional on $\mathbf{z}_i$, $\mathbf{y}_i$'s are independent. Also conditional on $\mathbf{z}_i$ all the components of $\mathbf{y}_i$ are independent among themselves. Thus we can write

$$\mathbf{y}_i = \mathbf{z}_i + \boldsymbol{\eta}_i \quad (3-29)$$

where $\boldsymbol{\eta}_i = (\eta_{i1}, \dots, \eta_{iq})^T$ is a multivariate normal random variate with mean zero and variance $\mathbf{V} = \text{diag}(\sigma_1^2, \dots, \sigma_q^2)$. Thus the components of $\mathbf{y}_i$ are kept independent conditional on $\mathbf{z}_i$ but each component is allowed to have a different variance. In this way we allow for any possible heteroscedasticity in the components of $\mathbf{y}_i$. This gives enormous flexibility in the modelling. In the regular cases we put a variance covariance matrix on the variance of $\boldsymbol{\eta}_i$ and assign an inverse Wishart prior on it. But it largely restrains the modelling, as by putting an inverse Wishart distribution we are effectively giving same degrees of freedom for all the variance components. Also, we establish the dependence between different components of the response vector $\mathbf{y}_i$ by introducing the latent variables $\mathbf{z}_i$ and connect $\mathbf{z}_i$ with $\mathbf{f}(\mathbf{x}_i) = (f_1(\mathbf{x}_i), \dots, f_q(\mathbf{x}_i))^T$ by $\mathbf{z}_i = \mathbf{f}(\mathbf{x}_i) + \boldsymbol{\delta}_i$, $\boldsymbol{\delta}_i = (\delta_{i1}, \dots, \delta_{iq})^T$ being the vector of the residual random effects. Assuming that $\mathbf{f}$ is generated from the product space of q RKHS's $\mathcal{H}_K$ with the kernel function $K(\cdot, \cdot)$, the representation theorem of Kimeldorf and Wahba (Equation 1-21) leads to the representation given in Equation 3-30.

$$\mathbf{z}_i = \mathbf{K}_i^0 \boldsymbol{\beta} + \boldsymbol{\delta}_i \quad (3-30)$$

where $K_i^0 = I_q \otimes K_i^T$, $\beta = (\beta_1^T, \dots, \beta_q^T)^T$ and $\delta_i \stackrel{iid}{\sim} N_q(0, \Sigma)$. The δ_i introduce dependence in the components of z_i , which in turn creates association among the components of y_i .

We assign priors on the unknown parameters $\beta, \Lambda, z, V, \Sigma, \theta$ as in Equations 3-31 to 3-35.

$$z_i | \beta, \Sigma, \theta \stackrel{iid}{\sim} N_q(K_i^0 \beta, \Sigma) \quad (3-31)$$

$$\beta_j | \Lambda_j \stackrel{iid}{\sim} N_{n+1}(0, \Lambda_j^{-1}); \Lambda_j = \text{diag}(\lambda_{0j}, \dots, \lambda_{nj}) \quad (3-32)$$

$$\Sigma \sim \text{IW}(\psi, Q) \quad (3-33)$$

$$\sigma_j^2 \stackrel{iid}{\sim} \text{IG}(\gamma_1, \gamma_2) \quad (3-34)$$

$$\theta \sim U(a_L, a_U), \quad \lambda_{ij} \stackrel{iid}{\sim} \text{Gamma}(c, d) \quad (3-35)$$

where $j = 1, \dots, q$, $i = 1, \dots, n$.

Let $\Lambda = \text{diag}(\Lambda_1, \dots, \Lambda_q)$. An inverse Wishart distribution with parameters ψ and Q for a random variable S has distribution function proportional to $f(S) \propto |S|^{-\frac{1}{2}(\psi+p+1)} \exp[-\frac{1}{2}\text{tr}(S^{-1}Q)]$. We refer to it as $\text{IW}(\psi, Q)$. The joint posterior is given by Equation 3-36.

$$\begin{aligned} \pi(\beta, \Lambda, z, V, \Sigma, \theta | y) &\propto \exp\left(-\frac{1}{2} \sum_{i=1}^n (y_i - z_i)^T V^{-1} (y_i - z_i)\right) \\ &\times \frac{1}{|\Sigma|^{n/2}} \exp\left(-\frac{1}{2} \sum_{i=1}^n (z_i - K_i^0 \beta)^T \Sigma^{-1} (z_i - K_i^0 \beta)\right) \\ &\times |\Sigma|^{-\frac{1}{2}(\psi+q+1)} \exp\left[-\frac{1}{2}\text{tr}(\Sigma^{-1}Q)\right] \\ &\times \frac{1}{|\Lambda|^{-1/2}} \exp\left(-\frac{1}{2}\beta' \Lambda \beta\right) \\ &\times \prod_{j=1}^q \exp\left(-\frac{\gamma_2}{\sigma_j^2}\right) (\sigma_j^2)^{-\gamma_1-1} \\ &\times \prod_{i=1}^n \prod_{j=1}^q \exp(-d\lambda_{ij}) \lambda_{ij}^{c-1}. \end{aligned} \quad (3-36)$$

The posterior distribution is analytically intractable. Once again we use MCMC techniques such as the Gibbs sampling and MH algorithm which require generating samples from the full conditionals of the different parameters given the remaining parameters and the data. We list the conditional distributions as follows:

$$(i) \beta | \Lambda, \mathbf{z}, \mathbf{V}, \Sigma, \theta, \mathbf{y} \sim N_{q(n+1)}(\mu_\beta^*, \mathbf{V}_\beta^*),$$

$$\text{where } \mu_\beta^* = \mathbf{V}_\beta^* (\sum_{i=1}^n \mathbf{K}_i^{0T} \Sigma^{-1} \mathbf{z}_i), \mathbf{V}_\beta^* = (\sum_{i=1}^n \mathbf{K}_i^{0T} \Sigma^{-1} \mathbf{K}_i^0 + \mathbf{V})^{-1}.$$

$$(ii) \Sigma | \beta, \Lambda, \mathbf{z}, \mathbf{V}, \theta, \mathbf{y} \sim \text{IW}(\psi^*, \mathbf{Q}^*),$$

$$\text{where } \psi^* = n + \psi, \mathbf{Q}^* = \mathbf{Q} + \sum_{i=1}^n (\mathbf{z}_i - \mathbf{K}_i^0 \beta)(\mathbf{z}_i - \mathbf{K}_i^0 \beta)^T.$$

$$(iii) \sigma_j^2 | \beta, \Lambda, \mathbf{z}, \Sigma, \theta, \mathbf{y} \stackrel{\text{ind}}{\sim} \text{IG}(\gamma_1^*, \gamma_2^*), j = 1, \dots, q,$$

$$\text{where } \gamma_1^* = n/2 + \gamma_1, \gamma_2^* = \sum_{i=1}^n \frac{(y_{ij} - z_{ij})^2}{2} + \gamma_2.$$

$$(iv) \lambda_{ij} | \beta, \mathbf{V}, \mathbf{z}, \Sigma, \theta, \mathbf{y} \stackrel{\text{ind}}{\sim} \text{Gamma}(c^*, d^*), j = 1, \dots, q; i = 1, \dots, n.$$

$$\text{where } c^* = c + 1/2 \text{ and } d^* = \frac{\beta_{ij}^2}{2} + d.$$

$$(v) \mathbf{z}_i | \Lambda, \beta, \mathbf{V}, \Sigma, \theta, \mathbf{y} \stackrel{\text{ind}}{\sim} N_q(\mu_{z_i}^*, \Sigma_{z_i}^*),$$

$$\text{where } \mu_{z_i}^* = \Sigma_{z_i}^* (\mathbf{V}^{-1} \mathbf{y}_i + \Sigma^{-1} \mathbf{K}_i^0 \beta); \Sigma_{z_i}^* = (\mathbf{V}^{-1} + \Sigma^{-1})^{-1}.$$

$$(vi) p(\theta | \Lambda, \beta, \mathbf{V}, \Sigma, \mathbf{z}, \mathbf{y}) \propto \exp \left[-\frac{1}{2} \sum_{i=1}^n (\mathbf{z}_i - \mathbf{K}_i^0 \beta)^T \Sigma^{-1} (\mathbf{z}_i - \mathbf{K}_i^0 \beta) \right].$$

The conditional distributions given in (i)-(v) are standard and it is easy to generate samples from them. The conditional distribution in (vi) is not standard and we need to employ the MH algorithm. If θ is the current value, draw a candidate value θ^* from $U(a_L, a_U)$. Accept the θ^* as a new value of θ with acceptance probability

$$\alpha = \min \left\{ 1, \frac{p(\theta^* | \Lambda, \beta, \mathbf{V}, \Sigma, \mathbf{z}, \mathbf{y})}{p(\theta | \Lambda, \beta, \mathbf{V}, \Sigma, \mathbf{z}, \mathbf{y})} \right\}. \quad (3-37)$$

3.5 Hierarchical Bayes SVM for Multivariate Regression

In this section we extend the idea of support vector machine and the Vapnik's ϵ -sensitive loss function to the multivariate case. We define a loss function which is very similar to Vapnik's ϵ -insensitive loss function in multivariate set up as follows.

Like in the case of RVM in Section 3.4 we introduce latent variables $\mathbf{z}_1, \dots, \mathbf{z}_n$ and assume that conditional on $\mathbf{z}_i$ components of $\mathbf{y}_i$ are independent. We generalize

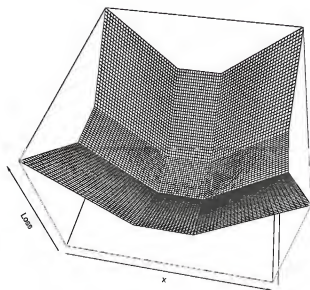


Figure 3-2: Our ϵ -insensitive loss function when $q = 2$ and $\rho_1 = \rho_2 = 1$ ($\epsilon = 1$).

Vapnik's ϵ -insensitive loss in a multivariate setup (Figure 3-2) by the Equation 3-38.

$$L(\mathbf{y}_i, \mathbf{z}_i) = \|\mathbf{y}_i - \mathbf{z}_i\|_\epsilon = \sum_{j=1}^q \rho_j |y_{ij} - z_{ij}|_\epsilon. \quad (3-38)$$

The difference between the multivariate SVM and RVM is that in the multivariate SVM, our loss function involves the truncated L^1 norm instead of the squared norm as in RVM. In one dimension, it is the L_1 loss instead of the L_2 loss. The dependence among the components of $\mathbf{y}_i$ is established by the dependence in the components of $\mathbf{z}_i$. The likelihood of $\mathbf{y}_i$ conditional on $\mathbf{z}_i$ corresponding to the loss (Equation 3-38) is given by

$$p(\mathbf{y}_i | \mathbf{z}_i) \propto \exp \{-\|\mathbf{y}_i - \mathbf{z}_i\|_\epsilon\}. \quad (3-39)$$

As in the previous section, we assume that the underlying function $\mathbf{f}$ is generated from the product space of q RKHS's and use the representation theorem (Equation 1-21) to connect the latent variables with $\mathbf{f}$ as in Equation 3-30 of the previous section. We specify hierarchical priors as in Equations 3-31 to 3-33, and Equation 3-35.

In addition, we assign priors for the scale parameter ρ_j and ϵ as follows:

$$\rho_j \stackrel{iid}{\sim} U(r_L, r_U) \text{ and } \epsilon \sim \text{Beta}(k_1, k_2), \quad j = 1, \dots, q. \quad (3-40)$$

The resulting joint posterior is given by Equation 3-41.

$$\begin{aligned} \pi(\beta, \Lambda, \mathbf{z}, \Sigma, \theta, \rho, \epsilon | \mathbf{y}) &\propto \exp \left(- \sum_{i=1}^n \| \mathbf{y}_i - \mathbf{z}_i \|_{\epsilon} \right) \\ &\times \frac{1}{|\Sigma|^{n/2}} \exp \left(- \frac{1}{2} \sum_{i=1}^n (\mathbf{z}_i - \mathbf{K}_i^0 \beta)^T \Sigma^{-1} (\mathbf{z}_i - \mathbf{K}_i^0 \beta) \right) \\ &\times |\Sigma|^{-\frac{1}{2}(\psi+q+1)} \exp \left[- \frac{1}{2} \text{tr}(\Sigma^{-1} \mathbf{Q}) \right] \\ &\times \frac{1}{|\Lambda|^{-1/2}} \exp \left(- \frac{1}{2} \beta' \Lambda \beta \right) \\ &\times \prod_{i=1}^n \prod_{j=1}^q \exp(-d\lambda_{ij}) \lambda_{ij}^{\epsilon-1} \\ &\times \epsilon^{k_1-1} (1-\epsilon)^{k_2-1}. \end{aligned} \quad (3-41)$$

Our analysis is once again based on the MCMC technique. The conditional distributions are derived as before. Conditional posterior distribution of $\beta, \Sigma, \Lambda, \theta$ are the same as (i), (ii), (iv), and (vi) of Section 3.4. Change in the loss function results in a change in the conditional distribution of $\mathbf{y}_i$ given $\mathbf{z}_i$. SVM model has the scale parameter $\rho = (\rho_1, \dots, \rho_q)^T$ instead of σ_j in case of RVM. The conditional posteriors for ρ_j and $\mathbf{z}_i$ replaces (iii) and (v) in the previous section as:

$$\begin{aligned} \text{(iii)} \quad p(\rho_j | \beta, \Lambda, \mathbf{z}, \Sigma, \theta, \epsilon, \mathbf{y}) &\propto \frac{\rho_j^{\theta}}{(1+\epsilon\rho_j)^{\theta}} \exp(-\rho_j \sum_{i=1}^n |y_{ij} - z_{ij}|_{\epsilon}), \quad j = 1, \dots, q. \\ \text{(v)} \quad p(\mathbf{z}_i | \beta, \Lambda, \Sigma, \theta, \rho, \epsilon, \mathbf{y}) &\propto \exp(-\|\mathbf{y}_i - \mathbf{z}_i\|_{\epsilon}) \times \exp \left[- \frac{1}{2} (\mathbf{z}_i - \mathbf{K}_i^0 \beta)^T \Sigma^{-1} (\mathbf{z}_i - \mathbf{K}_i^0 \beta) \right] \\ &i = 1, \dots, n. \end{aligned}$$

The conditional posterior for ϵ is given by

$$(vi) p(\epsilon|\beta, \Lambda, \mathbf{z}, \Sigma, \theta, \rho, \mathbf{y}) \propto \frac{1}{\prod_{j=1}^n (1 + \epsilon \rho_j)^n} \exp \left(-\epsilon \sum_{i=1}^n \sum_{j=1}^q \rho_j \mathbb{I}(|y_{ij} - z_{ij}| > \epsilon) \right) \times \epsilon^{k_1-1} (1 - \epsilon)^{k_2-1}.$$

Generating from the conditionals (i),(ii),and (iv) is easy. We generate samples from (iii) and (vi) using MH exactly in the same fashion as in Sections 3.3 and 3.4 respectively. The distribution (v) is not standard and is particularly difficult to generate from. We use MH algorithm with the proposal distribution that generates moves from the current state $\mathbf{z}_i$ to a new state $\mathbf{z}_i^*$ to be a multivariate Gaussian with mean equal to $\mathbf{K}_i^0 \beta$ and dispersion matrix Σ . The proposed updates are accepted with probability $\alpha_i = \min \left\{ 1, \frac{\exp(-\|\mathbf{y}_i - \mathbf{z}_i^*\|_\epsilon)}{\exp(-\|\mathbf{y}_i - \mathbf{z}_i\|_\epsilon)} \right\}$. Similarly MH is also employed to generate samples from (vi).

3.6 Prediction

As mentioned in the introduction, our main goal is to predict a new y_{new} when we have the corresponding covariates $\mathbf{x}_{new}$. For the prediction purpose we make use of the posterior predictive probability distribution of y_{new} given by Equation 3-42.

$$p(y_{new}|\mathbf{x}_{new}, \mathbf{y}) = \int_{\Theta} p(y_{new}|\mathbf{x}_{new}, \Theta, \mathbf{y}) \pi(\Theta|\mathbf{y}) d\Theta \quad (3-42)$$

where Θ is the vector of all model parameters. The new observation y_{new} is predicted by $\hat{y}_{new} = E[y_{new}|\mathbf{x}_{new}, \mathbf{y}]$. This integral is analytically intractable; so we use MCMC technique to evaluate it as follows:

Step 1. Generate M samples of the model parameter Θ from the posterior $\pi(\Theta|\mathbf{y})$.

Step 2. Generate M samples of $\mathbf{y}$ from the distribution $p(y_{new}|\mathbf{x}_{new}, \Theta_i, \mathbf{y})$, where

Θ_i is the i th sample of the model parameter and y_i^{gen} is the corresponding sample generated..

Step 3. Then $\hat{y}_{new} = \sum_{i=1}^M y_i^{gen} / M$.

3.7 Case Study

3.7.1 Application to Fluorescence Spectra

Research in fluorescence based optics suggests, that a less invasive measurement technique may be able to continuously monitor glucose levels in the body of diabetics. The data presented here is discussed in Spiegelman, Wikander, O'Neal, and Côté (2002). Optical spectra are collected from an experiment measuring photon counts for 101 wavelengths in the range 509-625 nm. Three replicate measurements on a particular glucose concentration in the range 0-160mg/dL in 20mg/dL increments are collected. So our response is that the glucose concentration and the covariates are the different photon counts corresponding to the $p = 101$ wavelengths. The covariates and the responses are centered and scaled around the column mean and column standard deviation. We randomly split our data set into training set, consisting of $n = 18$ samples and a test set, consisting of $m = 9$ samples. This is repeated 10 times. Our target is to predict accurately the glucose concentration in the body of the diabetics in the test set on the basis of the available photon counts. It is evident that the number of covariates p is far greater than n . We use the BRVM and BSVM models discussed in Section 3.2 and 3.3 for prediction of glucose concentration in a future observation. And compare the performance of our procedures with some of the classical procedures like partial least squares, principal component regression as well as classical support vector machine (CSVM). The performance is compared on the basis of average mean square errors of prediction (MSEP) over all 10 different splits of data sets.

We generate a MCMC sample of 100,000 with the first half as the burn in. Then out of the last half we select the 5th sample to reduce the correlation among the generated samples. To avoid the issue of multimodality and MCMC being stuck at one local mode we use 4 different starting points. Final prediction is obtained after pooling samples from the 4 chains.

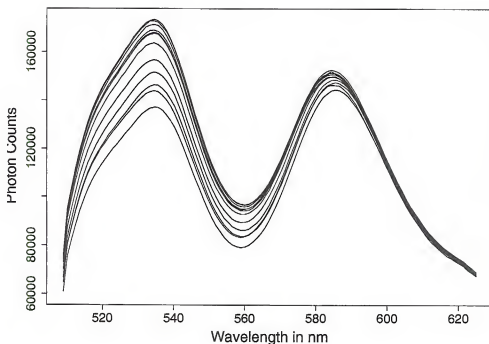


Figure 3-3: Optical spectra for 101 wavelengths on the bloodsugar data.

To make our models less sensitive to the choice of hyperparameters of the priors we have chosen near diffuse but proper priors. Near diffuse priors are proper priors but with large variance. Thus we can guarantee the propriety of the posterior and at the same time near diffuseness introduces some objectivity in our analysis. We have assigned a vague but proper prior to σ^2 . For λ we select the values of the hyperparameters so that the mean is kept very small around 0.001, but the variance is large. This choice of hyperparameter produces a near diffuse proper prior for β also. In all the examples in this chapter we have used the polynomial kernel. For the kernel parameter θ we give a discrete uniform prior $U\{1, 2, \dots, C\}$. In order to examine sensitivity in the choice of priors, we have considered several different combinations of near-diffuse but proper priors. The prediction error remains almost the same with all such choices. Here we report the results for two such choices (i) $\gamma_1 = 1, \gamma_2 = 10, c = 10^{-8}, d = 10^{-5}, C = 20$, and (ii) $\gamma_1 = 0.5, \gamma_2 = 1, c = 10^{-9}, d =$

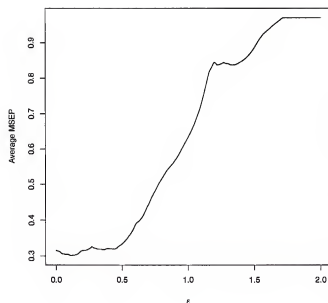


Figure 3-4: Average MSEP in the test set for different choices of ϵ in the bloodsugar data.

10^{-6} , $C = 10$. In BSVM additional parameters ϵ and ρ are drawn from a large support uniform prior. For ρ two choices of its hyperparameters are considered (i) $r_L = 0$, $r_U = 100$, and (ii) $r_L = 0$, $r_U = 50$. Choice of prior for ϵ is made such that the fitting is less sensitive to outliers. The ϵ parameter controls the width of the ϵ -insensitive zone, used to fit the training data. The value of ϵ can affect the number of support vectors used to construct the regression function. The bigger the ϵ , the fewer the support vectors selected. On the other hand, bigger ϵ -values results in more “flat” estimates. We have done a simple cross-validation (Figure 3-4) for finding out the range of ϵ where the SVM regression gives best results, and found out empirically that it works best in the range $(0, 0.5)$. Hence we have assigned a $\text{Beta}(k_1, k_2)$ distribution to ϵ . We have tried several combinations of (k_1, k_2) and the results are fairly similar. Here we report our results using (i) $k_1 = k_2 = 1$, i.e. we put a uniform $U(0, 1)$ prior on ϵ , and (ii) $k_1 = 1, k_2 = 5$. Law and Kwok (2001) proposed a data dependent prior on ϵ , but sampling from the posterior under their prior becomes much more complicated. It

may be noted that a better prediction accuracy can be attained by choosing a tight prior properly centered. However if this does not hold, the prediction will be highly inaccurate. The near-diffuse priors offers protection against this, and introduces some objectivity in our procedure.

We have used the *svm()* function in the R routine to fit the CSVM models. We have used $\epsilon = 0.1$ and $\theta = 2$, which is the default choice and also gave the best prediction result. For the regularization parameter λ , we used the default setting, i.e. $\lambda = 1$. For computing the PLS and PCR we have used the *mvr()* function in R with all of its default parameters. This parameter choice is also used in the next two examples.

Table 4-1 gives a comparative performance of various methods along with our two methods proposed earlier for prediction of blood glucose concentration on the basis of average MSEP. Both multiple smoothing parameters and single smoothing parameter is used. Results from the models with single smoothing parameter are marked by a “*”. It is clear that our RKHS based Bayesian RVM and SVM perform much better than all other methods like PLS, PCR which are often used by the chemometricians. Also Bayesian RVM and SVM leads to a better prediction than the classical SVM. Introduction of multiple smoothing parameters also introduces sparsity in our model, as the β_i 's are then controlled individually. A number of these will then be shrunk to zero introducing the sparsity. Mallick et al. (2005) has also provided a nice comparison of benefits of multiple smoothing parameter over a single smoothing parameter in a Bayesian SVM classification model. Introducing sparsity in this way also draws similarity to the Automatic Relevance Determination (ARD) models of Neal (1996) and MacKay (1994). In terms of performance, with multiple smoothing parameters an improvement of 10% to 30% in prediction accuracy is made over the single smoothing parameter. Apart from the improvement in prediction accuracy, it enhances the flexibility of the model as well. In particular, Bayesian

Table 3-1: Average mean squared errors of prediction (MSEP) on the 9 blood sample in the test set. Results from models with single smoothing parameters are marked by a *.

Method	Average MSEP
PLS	0.168
PCR	0.156
CSVM	0.265
Hyperparameter Choice (i)	
BRVM	0.054 0.063*
BSVM	0.042 0.057*
Hyperparameter Choice (ii)	
BRVM	0.060 0.082*
BSVM	0.044 0.065*
Empirical Bayes Models	
EBRVM	0.078 0.087*
EBSVM	0.065 0.092*

SVM with multiple smoothing parameters has the least average MSEP. Our model is not very sensitive to the choice of hyperparameters, as for both the choices we have nearly the same average MSEP. This is true as long as we stick to the class of near-diffuse but proper priors.

3.7.2 Application to NIR Spectroscopy Of Biscuit Doughs

The example studied here, arises from an experiment done to test the feasibility of near-infrared (NIR) spectroscopy to measure the composition of biscuit dough pieces formed from unbaked biscuits. The NIR spectrum of a sample is a continuous curve measured by modern scanning instruments. The information contained in this curve can be used to predict the chemical composition of the sample. A detailed description of the experiment can be obtained from Osborne, Fearn, Miller and Douglas (1984). The four constituents under investigation are: fat, sugar, flour and water. So our response is the calculated percentage of these four ingredients, $q = 4$. We made up two sets, one for training and a separate one for validation. The training set contains $n = 39$ samples, with sample 23 excluded from the original 40 as an outlier. A simple boxplot of the residuals from a PLS, PCR or MLR models confirm this fact. Further

the prediction set contains $m = 39$. Brown et al. (2001) analyzed this data using the Bayesian wavelet regression.

An NIR reflectance spectrum consists of 700 points measured from 1100 to 2498 nanometers (nm) in steps of 2 nm is available for each dough piece. Brown et al. (2001), reduced the number of spectral points by removing first 140 and last 49 wavelengths which are thought to contain little information. The selected wavelengths range from 1380 nm to 2400 nm, over which they took every other point, thus increasing the gap to 4 nm. So the number of spectral points are finally reduced to $p = 256$. In their case (Brown et al., 2001), this is done solely to save computational time, whereas in our SVM and RVM, the size of p does not really matter as by the representation theorem, we reduce the dimension from p to n . But to compare with their outcomes, we chose to use the same 256 spectral points.

Our aim is to predict the response values (composition of biscuit dough) from the spectral data for future samples where the responses are unknown but the spectral data are easily obtained. Several models are considered and compared in this context: (i) Stepwise multiple linear regression (MLR) (Osborne et al., 1984). (ii) Bayesian decision theory approach of Brown et al., (1999). (iii) Partial least squares regression (PLS) (iv) Principal components regression (PCR). (v) Bayesian wavelet regression (BWR) of Brown et al., (2001). (vi) Classical SVM (CSVM). Methods (i)-(v) are discussed in details in Brown et al, 2001. As the response is multivariate ($q=4$), hence we apply our (vii) multivariate Bayesian RVM (MBRVM) of Section 3.4 and (viii) multivariate Bayesian SVM (MBSVM) of Section 3.5 to analyze this data. We also apply the (ix) Bayesian RVM (BRVM) of Section 3.3 and (x) Bayesian SVM (BSVM) of Section 3.4 separately on each of the constituents and compare their performance. We fit all our models using the polynomial kernel and multiple smoothing parameters. It is seen that in this case the polynomial kernel gives a better prediction, i.e. lower mean squared errors of prediction than the Gaussian kernel.

For the choice of hyper-parameters for θ , σ^2 , λ , ϵ , and ρ in model (ix) and (x), we use exactly the same near-diffuse but proper priors as in the blood sugar data example in the previous section. In the multivariate RVM and RVM models (vii) and (viii), choice of γ_1 , γ_2 , c , d and C are same as before. Along with that for the new parameter Σ which establishes the dependency among the components we assign: (i) $\psi = 5$, and $\mathbf{Q} = 10\mathbf{I}_4$, and (i) $\psi = 10$, and $\mathbf{Q} = \mathbf{I}_4$. In all models we generate MCMC sample of 100,000 with a burn-in of first 50,000. Every 5th sample is retained in the next 50,000 samples to reduce the auto-correlation. To avoid any potential problem due to multimodality of the posterior we use 4 different starting points. Final predicted value is obtained after pulling samples from all 4 chains.

From Table 3-2, we see that the Bayesian SVM and RVM (multivariate and univariate) reduce the mean squared errors of prediction (MSEP) considerably, compared to other standard methods. Bayesian RVM and SVM also outperforms classical SVM by a considerable margin in predicting most of the components. Although our response is multivariate, we still notice that when we apply Bayesian RVM and SVM separately for each component, the results are quite impressive. For components like sugar and flour, it predicts better than BWR which is based on treating the response as multivariate. Similarly, we see that the MBSVM and MBRVM work even better than their univariate counterparts. Finally MBRVM comes out as the best of all the models discussed, in predicting the composition of the biscuit dough pieces in the validation set. This is true for both the choices of hyperparameters. We can see that even by changing the prior parameters, the prediction error remains almost the same which validates our claim that due to our choice of near-diffuse priors, our models are not very sensitive. Indeed our methods are not tuned just to these particular datasets but can be successfully applied to other data sets with the same of "large p small n " structure.

Table 3-2: Mean squared errors of prediction on the 39 biscuit dough pieces in the validation set.

Method	Fat	Sugar	Flour	Water
Stepwise MLR	0.044	1.188	0.722	0.221
Decision theory	0.076	0.566	0.265	0.176
PLS	0.151	0.583	0.375	0.105
PCR	0.160	0.614	0.388	0.106
BWR	0.063	0.449	0.348	0.050
CSVM	0.072	0.730	0.386	0.114
Hyperparameter Choice (i)				
BRVM	0.071	0.414	0.315	0.083
BSVM	0.072	0.441	0.316	0.091
MBRVM	0.057	0.314	0.252	0.031
MBSVM	0.062	0.339	0.229	0.045
Hyperparameter Choice (ii)				
BRVM	0.067	0.431	0.302	0.089
BSVM	0.071	0.437	0.311	0.091
MBRVM	0.057	0.320	0.236	0.038
MBSVM	0.065	0.343	0.232	0.050
Empirical Bayes Models				
EBRVM	0.089	0.596	0.324	0.105
EBSVM	0.084	0.420	0.321	0.097
EMBRVM	0.065	0.439	0.309	0.075
EMBSVM	0.074	0.477	0.341	0.056

Table 3-3: Mean squared errors of prediction on the 39 biscuit dough pieces in the validation set when the sample 23 is included in the training set.

Method	Fat	Sugar	Flour	Water
Hyperparameter Choice (i)				
BRVM	0.082	3.071	2.473	0.224
BSVM	0.102	0.846	0.465	0.116
MBRVM	0.089	2.761	2.842	0.109
MBSVM	0.089	0.675	0.396	0.101
Hyperparameter Choice (ii)				
BRVM	0.094	3.121	2.331	0.261
BSVM	0.104	0.832	0.422	0.126
MBRVM	0.097	3.111	2.689	0.122
MBSVM	0.081	0.613	0.412	0.132
Empirical Bayes Models				
EBRVM	0.147	6.6254	3.712	0.219
EBSVM	0.121	0.901	0.417	0.182
EMBRVM	0.161	5.325	3.982	0.176
EMBSVM	0.097	0.742	0.369	0.153

The sample 23 is excluded from the training set in the original analysis by Osborne et al. (1984) and also by Brown et al. (2001) as an outlier. As our main objective was to compare our models with their's, we too decided to exclude sample 23. However when sample 23 is included the result we obtain is listed in Table 3-3. It is clear from the table that by adding the outlier sample 23 our prediction accuracy gets diminished by a large margin specially in the sugar and flour component. However we may note that our BSVM and MBSVM models which are based on robust ϵ -insensitive loss remains much less affected, in contrast to the BRVM and MBRVM which are based on squared error loss function. This indeed supports our claim that Vapnik's loss makes the fitting less sensitive to the outliers.

3.7.3 Simulation Study

Using the biscuit dough dataset as the prototype and hyperparameter choice (i) we simulate a realistic dataset. Posterior means of all model parameters of our multiple shrinkage MBRVM model is used as the values of all model parameters for generating data. We perform simulations to generate the multivariate response $\mathbf{Y}$, using the MBRVM model. Thus we get a dataset with 78 samples, which we randomly split into half for the purpose of creating training and test sets.

We apply all our univariate and multivariate models discussed in the previous sections to this simulated dataset. In Table 3-4 we report our findings. Since MBRVM is used to generate the dataset, ideally we should get the lowest prediction error when we use MBRVM. Our findings in Table 3-4, also indicates that we get the lowest MSEP when we use the MBRVM. Also we notice that in spite of the fact that the hyperparameter choice (i) is used for simulation, when we do our prediction we use both the choices of hyperparameters. In both the choices we get similar prediction accuracy which tells us that misspecification of priors does not affect our prediction accuracy much, as long as we remain in the class of near-diffuse priors. When we use MBSVM, although the results are not very impressive compared to MBRVM yet it is

Table 3-4: Mean squared errors of prediction in the simulated dataset.

Method	Fat	Sugar	Flour	Water
Hyperparameter Choice (i)				
BRVM	0.051	0.489	0.477	0.120
BSVM	0.060	0.474	0.501	0.190
MBRVM	0.033	0.223	0.103	0.024
MBSVM	0.062	0.398	0.326	0.088
Hyperparameter Choice (ii)				
BRVM	0.059	0.463	0.511	0.089
BSVM	0.068	0.481	0.496	0.091
MBRVM	0.041	0.320	0.215	0.068
MBSVM	0.061	0.413	0.358	0.090
Empirical Bayes Models				
EBRVM	0.061	0.345	0.451	0.100
EBSVM	0.079	0.302	0.509	0.108
EMBRVM	0.055	0.344	0.236	0.090
EMBSVM	0.080	0.369	0.375	0.089

also not very far off. It signifies the robustness property of the MBSVM inherent from Vapnik's ϵ -insensitive loss function. The univariate BRVM and BSVM also provide competitive results but they are definitely poorer than their multivariate counterparts due to the simple fact that they cannot take into account the dependence among the components of the responses $\mathbf{Y}$.

3.8 Empirical Bayes Support Vector Machine And Relevance Vector Machine

In the previous sections, we have introduced hierarchical Bayesian SVM and RVM for both univariate and multivariate setups for "large p small n " type regression. Hyperparameters for θ , λ , ρ must be specified to carry out the analysis. Utmost care should be taken for the choice of these prior parameters as they make our model very sensitive, and hence the objectivity of their application becomes widely debated. We tried to contain the issue of choice prior parameters by choosing near-diffuse proper priors. In this section, we will provide an alternative empirical Bayes analysis for SVM and RVM models, which is considered to be an approximation of a full Bayes approach.

In the empirical Bayes RVM in univariate setup (EBRVM), we do not assume any prior distribution on the kernel parameter θ and the shrinkage parameter λ . Rather we estimate θ and λ by maximizing the joint marginal posterior, which is given by Equation 3-43.

$$\begin{aligned} \pi(\theta, \lambda | \mathbf{y}) &\propto \int_{\sigma^2} \int_{\beta} \pi(\beta, \lambda, \sigma^2, \theta | \mathbf{y}) d\beta d\sigma^2 \\ &\propto \frac{\prod_{i=1}^n \lambda_i^{1/2}}{|K_0^T K_0 + D|^{1/2}} \frac{\Gamma(\gamma_1^*)}{(\gamma_2^*)^{\gamma_1}} \end{aligned} \quad (3-43)$$

where $\gamma_1^* = \frac{2n+1}{2} + \gamma_1$, $\gamma_2^* = \frac{\mathbf{y}^T A \mathbf{y}}{2} + \gamma_2$, and $A = I_n - K_0(K_0^T K_0 + D)^{-1} K_0^T$. When we assume only one tuning parameter $\lambda = (\lambda_0, \lambda, \dots, \lambda)$, $\prod_{i=1}^n \lambda_i^{1/2}$ in Equation 3-43 simplifies to $\lambda^{n/2}$.

Similarly in the empirical Bayes SVM, in the univariate case (EBSVM), we estimate θ , λ , and ρ by maximizing the marginal posterior given in Equation 3-44.

$$\begin{aligned}\pi(\theta, \lambda, \rho | \mathbf{y}) &\propto \int_{\sigma^2} \int_{\mathbf{z}} \int_{\beta} \pi(\beta, \lambda, \sigma^2, \theta, \rho | \mathbf{y}) d\beta d\sigma^2 d\mathbf{z} de \\ &\propto \int_{\epsilon} \int_{\mathbf{z}} g(\lambda, \rho, \theta) \epsilon^{k_1-1} (1-\epsilon)^{k_2-1} \exp\left(-\rho \sum_{i=1}^n |y_i - z_i|_{\epsilon}\right) d\mathbf{z} d\epsilon\end{aligned}\quad (3-44)$$

where $g(\lambda, \rho, \theta) = \frac{\prod_{i=1}^n \lambda_i^{1/2}}{|K_0^T K_0 + D|^{1/2}} \frac{\rho^n \Gamma \gamma_i^*}{(\gamma_2^*)^{1/2}}$ when we have multiple smoothing parameters and $g(\lambda, \rho, \theta) = \frac{\lambda^{n/2}}{|K_0^T K_0 + D|^{1/2}} \frac{\rho^n \Gamma \gamma^*}{(\gamma_2^*)^{1/2}}$ when we have single smoothing parameter. Simulated maximum likelihood method (McCulloch, 1997) is used to maximize the above posterior as follows:

Step 1. Start with an initial value of λ , ρ , and θ .

Step 2. Generate $(\epsilon^{(i)}, z^{(i)})$, $i = 1, \dots, m$. from $h(\epsilon)$ & $g(\mathbf{z}|\epsilon)$.

Step 3. Maximize $\Delta(\lambda, \rho, \theta) = \sum_{i=1}^m g(\lambda, \rho, \theta | z^{(i)}, \epsilon^{(i)})$ to get $\lambda^*, \rho^*, \theta^*$.

Step 4. $\lambda = \lambda^*$, $\rho = \rho^*$, $\theta = \theta^*$, go to Step 1.

Increase m as we move along.

The empirical Bayes analysis for the multivariate RVM (EMBRVM) is more complicated and computation intensive. We get the estimates of λ , and θ by maximizing the marginal posterior in Equation 3-45.

$$\begin{aligned}\pi(\theta, \lambda | \mathbf{y}) &\propto \int_{\beta} \int_{\Lambda} \int_{\Sigma} \int_{\mathbf{V}} \pi(\beta, \Lambda, \Sigma, \mathbf{V}, \mathbf{z}, \theta | \mathbf{y}) d\mathbf{V} d\Sigma d\beta d\mathbf{z} \\ &\propto \int_{\mathbf{z}} \int_{\beta} \frac{|\Lambda|^{1/2} \exp\left(-\frac{\beta^T \Lambda \beta}{2}\right) \prod_{j=1}^q A_j}{|B|^{(n+\psi)/2}} d\beta d\mathbf{z}\end{aligned}\quad (3-45)$$

where $A_j = \frac{\Gamma(n/2 + \gamma_j)}{\gamma_2 + \sum_{i=1}^n (y_{ij} - z_{ij})^2/2}$, and $B = Q + \sum_{i=1}^n (z_i - K_i^0 \beta)(z_i - K_i^0 \beta)^T$.

In the case of multivariate SVM, the empirical Bayes analysis (EMBSVM) is carried out by estimating λ , ρ , and θ by maximizing the marginal posterior in

Equation 3-36.

$$\pi(\theta, \Lambda, \rho | y) \propto \int_{\epsilon} \int_{\mathbf{z}} \int_{\beta} \int_{\Sigma} \pi(\beta, \Lambda, \Sigma, \mathbf{z}, \theta, \rho, \epsilon | y) d\Sigma d\beta dz d\epsilon \quad (3-46)$$

To maximize Equations 3-45 and 3-46 the simulated maximum likelihood method is used in a similar manner as in the univariate cases. After computing the estimated values of θ , ρ , and Λ , they are plugged in the posterior predictive distributions to find the final predicted value. The empirical Bayes method helps us to avoid the problem of specifying priors on θ , ρ , and Λ , but they are highly computation intensive and slow.

The mean squared errors of prediction on the blood sugar data set, biscuit dough data set, and the simulated data set in the empirical Bayes setup are provided in Tables 3-1 to 3-4. Comparing the results from these tables, we see that there is not much difference between the prediction accuracy of the empirical Bayes models, and our originally proposed full Bayesian models. However empirical Bayes method definitely has one advantage over the full Bayesian method in that case we can avoid specifying priors for many of the parameters, hence it is more adaptive to the needs of a particular data set. Yet the main problem that arises is from the maximization of the marginal likelihood as it is very common that we might be stuck to a local maxima. Multiple starting points must be used to overcome this fact. Although in all the datasets our empirical Bayes models gave inferior result than the hierarchical Bayes models, but they did better than the classical methods like PLS, PCR, MLR, and CSVM.

3.9 Concluding Remarks

The RKHS based SVM and RVM methodology turns out to be a strong contender for prediction. Specially, in the cases where we have a very large number of covariates but very small amount of data, its performance does not deteriorate with high input dimensionality. Unlike other machine learning methods, SVM and RVM do not

require an additional projection into sample space. Indeed, the dimension reduction is built automatically in the SVM and RVM methodology, as by Wahba's representation we are reducing the dimension from p to n .

In terms of computation speed for the most simple model, BRVM it is quite fast and takes about 5 hours to run 4 independent chains, each of 100,000 samples. The speed is also near the same for BSVM models. For MBRVM and MBSVM models as we need to invert $(n+1) \times (n+1)$ matrices, multiple times in each MCMC step, the progress is really slow, and takes about 10 to 15 hours to fit a model. From the trace plots, it is seen that after the first 50,000 runs, there is an acceptable amount of convergence in the chain. Empirical Bayes models are much faster as some of the parameters are estimated in advance, and then plugged in the posterior for inference. The biggest disadvantage of all the proposed methods when compared to other standard methods like PLS, PCR, and CSVM are in terms of speed. All these methods are "out of shelf" and excellent R routine exists that gives the result in no time when compared to our models. Regarding the feasibility of our methods for the large datasets, we want to point out that the sole application of our methods is for "large p small n " cases, i.e. when the number of observations is much smaller than the number of covariates. Instead when the sample size n exceeds p , Wahba representation blows up the dimension from p to n , and defeats the whole purpose of dimension reduction. Also the computation time, becomes unnecessarily large, as we need to invert a $(n+1) \times (n+1)$ matrix at each step.

The results from the two real life examples and one simulated dataset give strong credence to the view that our RKHS based Bayesian SVM and RVM help to build a much stronger learning algorithm than other straightforward methods like partial least square, principal component regression and stepwise multiple linear regression. It also performs much better than some very sophisticated methods like Bayesian wavelet regression of Brown et al. (2001) and Bayesian decision theory approach of

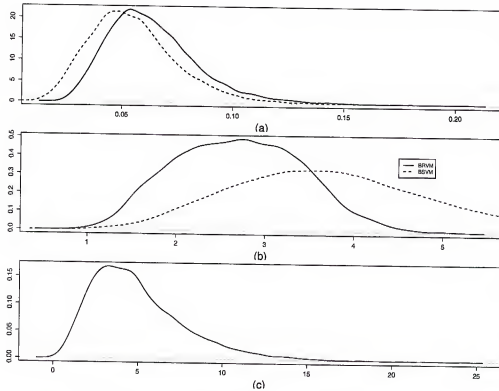


Figure 3-5: Blood sugar concentration data set.

(a) Distribution of average MSE under BRVM and BSVM. (b) Posterior distribution of the kernel parameter θ under BRVM and BSVM models. (c) Posterior distribution of ρ .

Brown et al. (1999). Bayesian SVM also has its virtues over the classical support vector machine by producing a richer class of models which give better prediction along with the unique ability to quantify the prediction error, i.e., now we can have the entire posterior distribution of MSE and can obtain a confidence interval (Figure 3-5 & Figure 3-6). Rather than having just a point predictor now we can have the full posterior predictive probability distribution of a future observation. The results from Table 3-2 and 3-3 indicate the success of our MBSVM method in handling outliers in the dataset. The MSE, specially in the sugar and flour component remains not very affected in the robust loss function based MBSVM model as compared to the MBRVM model which is based on the squared error loss function. This is good in terms of the robustness point of view.

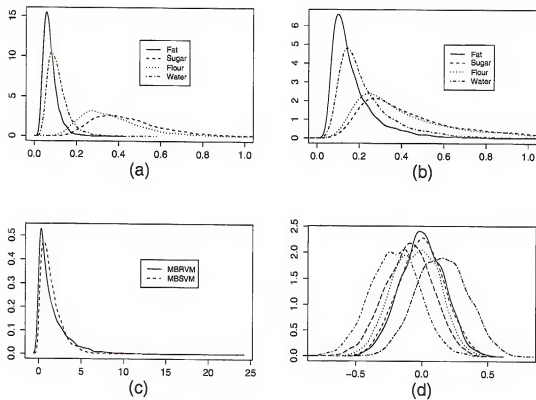


Figure 3-6: Biscuit dough composition data set.

The 4 components are fat, sugar, flour and water. (a) Distribution of MSEP for the 4 components of biscuit dough under MBRVM model. (b) Distribution of MSEP under MBSVM model. (c) Posterior distribution of the kernel parameter θ under MBRVM and MBSVM models. (d) Posterior distribution of the correlations under MBRVM models.

Bayesian SVM reduces the root mean squared errors of prediction by about 8% for the blood sugar data. For the biscuit data, BRVM performs marginally better than BSVM by reducing the mean squared errors of prediction in the range of 0.3% – 9%. The improvement of MBRVM over MBSVM is more pronounced, where the MSEP reduction ranges between 8% – 31%. Empirical Bayes RVM and SVM have their own advantages and disadvantages. In terms of prediction accuracy it comes close but is definitely inferior to the hierarchical Bayes models. However by adopting an empirical Bayes approach, one can avoid some prior elicitation problems. Choice of multiple smoothing parameters has a distinct advantage over the single smoothing parameter. Apart from the fact that multiple smoothing parameters provide independent shrinkage for the components of regression coefficients and result in sparsity, they also give better prediction. In blood sugar example we see that by using multiple smoothing parameters we reduce the MSEP by 10% – 30%. Thus, overall our recommendation is to use either hierarchical Bayes RVM or hierarchical Bayes SVM with multiple smoothing parameters for “large p small n ” regression problems.

As no method can lay claim to be universally best, we too cannot claim ours to be so. Therefore we would like to treat our models as a new probability based way of modeling and prediction in high-dimensional problems. Much of its success and limitations will be revealed as more and more applications are made on various new datasets.

CHAPTER 4

MULTICLASS CANCER DIAGNOSIS USING BAYESIAN KERNEL MACHINE MODELS

“... biology laboratories are now pumping out literally terabytes of information every month swamping themselves in a sea of indigestible data.”

— The Economist, April 2003.

The main objective of this chapter is to explore several machine learning techniques for multiclass classification of cancer tumors based on gene expression microarray covariates. We have introduced a full Bayesian model-based approach, specifically Bayesian multinomial logit model (BMLM) based on reproducing kernel Hilbert space (RKHS), as well as Bayesian support vector machines (BSVM) for multicategory classification. A stochastic search variable selection method is integrated with our models for simultaneously identifying the active genes from the dormant ones and using them for more accurate classification.

Cancer classification relies on the subjective interpretation of both clinical and histopathological information with an eye toward placing tumors in currently accepted categories based on the tissue of origin of the tumor. Current frameworks, however, are unable to discriminate among tumors with similar histopathologic features, which vary in clinical course and in response to treatment. There is an increasing interest in changing the basis of tumor classification from morphologic to molecular, using microarrays which provide expression measurements for thousands of genes simultaneously (Skena et al., 1995; DeRisi et al., 1997), a key goal being to perform classification via different expression patterns. Several studies using microarrays to profile colon, breast and other tumors have demonstrated the potential power of expression profiling for classification (Alon et al., 1999; Hedenfalk et al., 2001, Mallick et al., 2005).

However, these methods only handle binary classification problems. As there are hundred types of cancer (Hanahan and Weinberg, 2000) and even more subtypes, so the development of molecular classification procedure for multiclass data is essential. The problem is much more difficult than a binary classification problem, as there is only one positive or true class but many negative or other classes. Misclassification rates are much higher due to noise and variation of the expression data. Also due to lack of resources, the number of samples in each class is usually small (less than 100) which makes the problem more difficult. This is a problem of high dimension low sample size like in Chapter 3, but in the context of classification.

The support vector machine (SVM) has shown its popularity in microarray literature specially for binary classification problem. Usually there are two ways to handle multicategory problems using SVM: (i) Solve the multicategory problem by solving series of binary problems, as suggested by Dietterich and Bakiri (1995), and Allwein, Schapire, and Singer (2000), (ii) consider the classes at once as proposed by Vapnik (1995), Bredensteiner and Bennett (1999), and Crammer and Singer (2001). Constructing pairwise classifier or one-versus-rest classifiers is popular among the first approaches though they have a possible disadvantage of inflating the variance since smaller samples are used to learn each classifier. The second approach is a more algorithmic extension of the binary approach, without much connection with decision rule and uncertainty. Recently, Lee, Lin and Wahba (2004) proposed a coherent decision theoretic approach for multicategory support vector machine. They have used data adaptive tuning criterion for the smoothing parameters using generalized approximate cross validation (GACV) (Wahba et al., 2002).

In this chapter, we tackle the problem of cancer classification in the context of multiple tumor types, i.e the number of tumor classes are more than 2. We develop a full probabilistic model-based approach, specifically Bayesian support vector machines based on reproducing kernel Hilbert space (RKHS) (Lee et al.,

2004) and also a RKHS based Bayesian Multinomial logit model for multcategory classification. Here we will develop a hierarchical model where the unknown smoothing parameters will be interpreted as a shrinkage parameter (Denison et al., 2002). We will assign a prior distribution to it and obtain its posterior distribution via Bayesian paradigm. In this way, we not only obtain the point predictors but also the associated measures of uncertainty. Furthermore, we will extend the model to incorporate multiple smoothing parameters, leading to significant improvements in prediction for the example considered. The dimension reduction is built automatically in the BMLM and BSVM methodology; although an efficient variable selection method is required to enhance the performance of our model. We have proposed a stochastic search variable selection technique (George and McCulloch, 1993) using mixture priors for selecting differentially expressed genes.

We compare our procedure with some highly sophisticated methods like classical support vector machine (CSVM), neural network (NN), and random forest (RF). Almost all of these methods except random forest have an inherent weakness handling high dimensional data. Hence we have used Dudoit et al., (2002) "between square vs within square" (BWS/BSS) technique to order the full set of genes and then select only a few from the top to include in the models. A detailed description of the BWS/BSS criteria is given in Section 2.4.2 Chapter 2.

Section 4.1 of this chapter briefly describes the gene-expression microarray technique. Section 4.2, of this chapter introduces a RKHS-based Bayesian multinomial logit model. In Section 4.3, we develop a Bayesian support vector machine model for multcategory classification. A Bayesian gene selection scheme is provided in Section 4.4. In Section 4.5 we describe how to classify a new sample and identify the differentially expressed genes. In Section 6, we apply the general methodologies developed in Sections 4.2–4.5, to two different cancer classification problems. Finally, some concluding remarks are made in Section 4.7.

4.1 Gene Expression Microarrays

Proteins make up about 15% of the mass of an average person. Protein molecules are essential to us in an enormous variety of different ways. Much of the fabric of our body is constructed from protein molecules. Muscle, cartilage, ligaments, skin and hair - these are all mainly protein materials. In addition to these large scale structures that hold us together, smaller protein molecules play a vital role in keeping our body working properly. Haemoglobin, hormones, antibodies, and enzymes are all examples of these less obvious proteins. Genes control all cellular functions responsible for maintaining human health by serving as blueprints for the production of proteins in cells. Recent advances in cellular and molecular biology have shown that malfunctions in gene expression either cause or predispose humans to most diseases. Such malfunctions cause cells to produce inappropriate amounts or types of proteins. For example, the uncontrolled proliferation of cells characteristic of inflammatory diseases and cancer is the result of over-activation of specific genes and the subsequent over-production of proteins, such as cytokines and regulatory enzymes. Conversely, under-expression of critical genes and their protein products, such as tumor suppressors and growth factors, also may give rise to disease, including cancer and neurological disorders. The genes are again coded in the DNA or deoxyribonucleic acid. The production of proteins from the genes happens through two steps: (1) Transcription, and (2) Translation. Transcription is the process by which the enzyme RNA polymerase makes copies of genes in the form of mRNA or messenger RNA, allowing the information encoded in the DNA to be "expressed". After transcription the mRNA is used as a template to assemble all the chains of amino acids to form the protein this is known as translation. Gene expression measures the amount of transcribed mRNA in an organism. There are several methods for measuring gene expressions, and gene expression microarray is one of them. Figure 4-1 gives a schematic representation of production of proteins from genes.

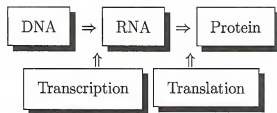


Figure 4-1: A schematic diagram of production of protein from DNA.

DNA microarrays measure the expression of a gene in a cell by measuring the hybridization, or matching, of DNA on a small glass slide, to the mRNA representation from the sample under study. Nucleotide sequence for a few thousand genes are printed on a glass slide. A target sample and reference sample is respectively labelled with red and green dyes. Then each are hybridized with the DNA on the slide. The slides are then passed through some scanning instruments and $\log(\text{red}/\text{green})$ intensities are measured by fluoroscopy. Positive values indicate higher expression in the target samples compared to the reference samples, and vice versa for negative values. The two microarray technologies available are: (a) cDNA arrays, and (b) oligonucleotide arrays. Both of them use the hybridization technique to quantify the transcribed mRNA, but they differ in how the DNA sequences are laid on the array and the length of these sequences. A gene expression microarray dataset is a large matrix whose each row represents an experiment and several thousands of columns represent individual genes. Figure 4-2 gives the picture of a typical DNA microarray data. Hence gene expression microarray data rightly fits into the “large p small n ” class of problems, as here also the number of covariates (p) or the genes are very large compared to the number of samples (n).

4.2 RKHS based Bayesian Multinomial logit model

Here we are interested in classification where the response is a categorical variable with more than two categories and the covariates are the gene expression microarray measurements. Let $\mathbf{t} = (t_1, \dots, t_n)$ indicates n observed response data. This t_i takes one of the J possible categories $(1, \dots, J)$, $J > 2$. Let $p_{ij} = P(t_i = j)$, for $i = 1, \dots, n$

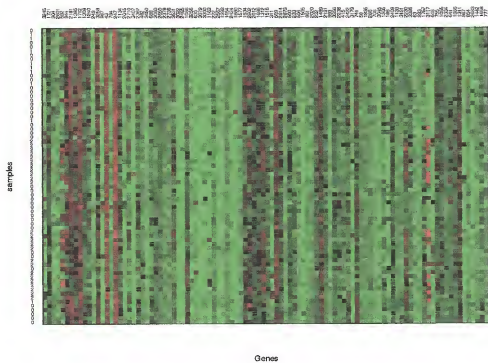


Figure 4-2: Heatmap of Golub leukemia DNA data.
Total 72 tissue samples, showing 3 types of leukemia 0 = B-ALL, 1 = T-ALL, 2 =
AML in rows and 100 randomly selected genes in columns.

and $j = 1, \dots, J$, denotes the probability that the i th observation falls into the j th category. We make an alternate representation of the response t_i by introducing a vector of indicator variable $\mathbf{y}_i = (y_{i1}, \dots, y_{iJ})^T$, where

$$y_{ij} = \begin{cases} 1 & \text{if } t_i = j \\ 0 & \text{otherwise} \end{cases} \quad (4-1)$$

$i = 1, \dots, n$ and $j = 1, \dots, J$.

The multinomial likelihood is

$$f(\mathbf{y}|\mathbf{p}) \propto \prod_{i=1}^n p_{i1}^{y_{i1}} \dots p_{iJ}^{y_{iJ}}. \quad (4-2)$$

These probabilities are related to a set of p gene expressions (covariates) $X_{n \times p}$ through a logistic link function and a hierarchical model as in Equation 4-3.

$$p_{ij} = \frac{\exp(z_{ij})}{\sum_{k=1}^J \exp(z_{ik})}. \quad (4-3)$$

We relate z_{ij} to $f_j(\mathbf{x}_i)$ by $z_{ij} = f_j(\mathbf{x}_i) + \epsilon_{ij}$, where ϵ_{ij} is the residual random effect. The f_j 's are unknown functions which connect the gene expressions with the tumor types. Taking the negative of the log of the multinomial likelihood (Equation 4-2) we get the corresponding loss function given by Equation 4-4.

$$L = - \sum_{i=1}^n \sum_{j=1}^J y_{ij} \log(p_{ij}). \quad (4-4)$$

This loss function is equivalent to the Kullback-Leibler (KL) directed divergence measure between y_{ij} and p_{ij} , given by Equations 4-5 and 4-6.

$$L_{KL} = \sum_{i=1}^n \sum_{j=1}^J y_{ij} \log(y_{ij}/p_{ij}) \quad (4-5)$$

$$= \sum_{i=1}^n \sum_{j=1}^J y_{ij} \log(y_{ij}) - \sum_{i=1}^n \sum_{j=1}^J y_{ij} \log(p_{ij}). \quad (4-6)$$

Maximizing the multinomial likelihood (Equation 4-2) will be equivalent to minimizing the KL loss function (Bernardo and Smith, 1994). To avoid overfitting we can add a penalty function. Casting the whole problem in the regularization framework as in Chapter 1 Section 1.4 and Chapter 3 Section 3.1 we have the following minimization problem:

$$\min_{f \in \mathcal{H}_K} \left[\sum_{i=1}^n L_{KL}(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}_K}^2 \right] \quad (4-7)$$

where $L_{KL}(\cdot)$ is the KL loss function described in Equation 4-5. $\|f\|_{\mathcal{H}_K}^2$ is the squared norm penalty functional, λ is the smoothing parameter, and $\mathbf{f} = (f_1, \dots, f_J)^T$ is the J -tuple classification function. We assume that f_j is generated from a reproducing kernel Hilbert space (RKHS) with a positive definite kernel function $K(\cdot, \cdot)$. Then the theory of RKHS as described in Kimeldorf and Wahba (197), and Wahba et al. (2002) leads to a representation of f_j as in Equation 4-8.

$$f_j(x_i) = \beta_{0j} + \sum_{k=1}^n \beta_{kj} K(x_i, x_k | \theta). \quad (4-8)$$

The non-linear predictor z_{ij} is thus treated as a random latent variable so that the model is now

$$z_{ij} = \beta_{0j} + \sum_{k=1}^n \beta_{kj} K(x_i, x_k | \theta) + \epsilon_{ij} = \mathbf{K}_i^T \boldsymbol{\beta}_j + \epsilon_{ij} \quad (4-9)$$

where the ϵ_{ij} are iid $N(0, \sigma^2)$, $\mathbf{K}_i = (1, K(x_i, x_1 | \theta)^T, \dots, K(x_i, x_n | \theta)^T)$, and $\boldsymbol{\beta}_j = (\beta_{0j}, \dots, \beta_{nj})^T$, $i = 1, \dots, n$. In practice only the first $J - 1$ elements of $\mathbf{y}_i$ are used in fitting the RKHS so that the problem is of full rank. So $z_{i,J}$ is set to 0 for all i for identifiability of the model. There are several possible choices of kernel functions, in this chapter, we used only the Gaussian kernel $K(x_i, x_j | \theta) = \exp\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\theta}\}$.

We assign hierarchical priors on the unknown parameters as described in Equations 4-10 to 4-13.

$$\beta_j | D_j, \sigma^2 \stackrel{iid}{\sim} N_{n+1}(0, \sigma^2 D_j^{-1}); D_j = \text{Diag}(\lambda_{0j}, \dots, \lambda_{nj}). \quad (4-10)$$

$$\sigma^2 \sim \text{IG}(\gamma_1, \gamma_2) \quad (4-11)$$

$$\theta \sim \text{U}(a_L, a_U) \quad (4-12)$$

$$\lambda_{ij} \stackrel{iid}{\sim} \text{Gamma}(c, d), \text{ where } j = 1, \dots, J-1, i = 1, \dots, n. \quad (4-13)$$

Denote $\lambda = (\lambda_{01}, \dots, \lambda_{n1}, \lambda_{02}, \dots, \lambda_{n2}, \dots, \lambda_{0(J-1)}, \dots, \lambda_{n(J-1)})^T$, and $\beta = (\beta_1^T, \dots, \beta_{J-1}^T)^T$, λ_{0j} are being fixed at small values to assign a large variance for the intercept terms β_{0j} , $j = \dots, J-1$. These λ is the vector of smoothing parameters. In the above formulation we have multiple smoothing parameters λ_{ij} 's, for the sake of simplicity we can also assign only one smoothing parameter as $\lambda_{ij} = \lambda$, for all $j = 1, \dots, J-1$ and $i = 1, \dots, n$.

The joint posterior is given by Equation 4-14.

$$\begin{aligned} \pi(z, \sigma^2, \beta, \lambda, \theta | y) &\propto \prod_{i=1}^n p_{i1}^{y_{i1}} \dots p_{iJ}^{y_{iJ}} \\ &\times \frac{\exp\{-\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^{J-1} (z_{ij} - K_i^T \beta_j)^2\}}{(\sigma^2)^{n(J-1)/2}} \\ &\times \frac{\exp\{-\frac{1}{2\sigma^2} \sum_{j=1}^{J-1} \beta_j^T D_j \beta_j\}}{(\sigma^2)^{(n+1)(J-1)/2} \prod_{j=1}^{J-1} |D_j^{-1}|^{1/2}} \\ &\times \exp(-\gamma_2/\sigma^2) (\sigma^2)^{-\gamma_1-1} \\ &\times \prod_{i=1}^n \prod_{j=1}^{J-1} \exp(-d\lambda_{ij}) (\lambda_{ij})^{c-1}. \end{aligned} \quad (4-14)$$

The posterior distribution is intractable so to generate samples from the posterior we use MCMC sampling techniques like Gibbs sampling (Gelfand and Smith, 1990) and Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953) as we did in the previous chapters. To generate from the joint posterior we use the full conditional distributions. The conditional distributions are listed as follows:

- (i) $\beta_j | \lambda, \mathbf{z}, \sigma^2, \theta, \mathbf{y} \sim N_{(n+1)}(\mu_{\beta_j}^*, V_{\beta_j}^*)$,
 where $\mu_{\beta_j}^* = V_{\beta_j}^* (\sum_{i=1}^n K_i z_{ij})$, $V_{\beta_j}^* = \sigma^2 \left(\sum_{i=1}^n K_i K_i^T + D_j \right)^{-1}$, for $j = 1, \dots, J-1$.
- (ii) $\sigma^2 | \beta, \lambda, \mathbf{z}, \theta, \mathbf{y} \stackrel{\text{ind}}{\sim} \text{IG}(\gamma_1^*, \gamma_2^*)$, where $\gamma_1^* = (J-1)(2n+1)/2 + \gamma_1$, and
 $\gamma_2^* = \frac{(\sum_{i=1}^n \sum_{j=1}^{J-1} (z_{ij} - K_i^T \beta_j)^2) + (\sum_{j=1}^{J-1} \beta_j^T D_j \beta_j)}{2} + \gamma_2$.
- (iii) $\lambda_{ij} | \beta, \mathbf{z}, \sigma^2, \theta, \mathbf{y} \stackrel{\text{ind}}{\sim} \text{Gamma}(c^*, d^*)$, $j = 1, \dots, J-1$; $i = 1, \dots, n$,
 where $c^* = c + 1/2$ and $d^* = \frac{\beta_{ij}^2}{2} + d$.
- (iv) $p(\theta | \lambda, \beta, \sigma^2, \mathbf{z}, \mathbf{y}) \propto \exp\{-\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^{J-1} (z_{ij} - K_i^T \beta_j)^2\} \times I(a_L < \theta < a_U)$,
 where $I(\cdot)$ is an indicator variable.
- (v) $p(\mathbf{z} | \theta, \lambda, \beta, \sigma^2, \mathbf{y}) \propto \prod_{i=1}^n p_{i1}^{y_{i1}} \dots p_{iJ}^{y_{iJ}} \times \frac{\exp\{-\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^{J-1} (z_{ij} - K_i^T \beta_j)^2\}}{(\sigma^2)^{n(J-1)/2}}$.

Generation from conditional distributions (i) to (iii) is easy as they each represent a standard probability distribution. The conditional for θ given by (iv) does not represent any particular distribution hence we devise a MH algorithm to sample from it. Let θ be the current state, then draw a candidate value θ^* from its prior $U(a_L, a_U)$. Accept the θ^* as a new value of θ with acceptance probability

$$\alpha = \min \left\{ 1, \frac{p(\theta^* | \lambda, \beta, \sigma^2, \mathbf{z}, \mathbf{y})}{p(\theta | \lambda, \beta, \sigma^2, \mathbf{z}, \mathbf{y})} \right\}. \quad (4-15)$$

The latent variables z_{ij} , $i = 1, \dots, n$; $j = 1, \dots, J-1$ are sampled using the data augmentation technique suggested by Albert and Chib (1993) as follows:

Step 1. Let the latent variable $\mathbf{z}_i = (z_{i1}, \dots, z_{iJ-1})$ be a vector corresponding to the i th subject.

Step 2. The relationship between y_{ij} and z_{ij} is given by Equation 4-16.

$$y_{ij} = \begin{cases} 0 & \text{if } \max_{1 \leq k \leq J-1} \{z_{ik}\} \leq 0 \\ j & \text{if } \max_{1 \leq k \leq J-1} \{z_{ik}\} > 0 \text{ and } z_{ij} = \max_{1 \leq k \leq J-1} \{z_{ik}\}. \end{cases} \quad (4-16)$$

Step 3. So $\mathbf{z}_{ij} \sim N(K_i^T \beta_j, \sigma^2)$ under the above constraint. If the i th subject belongs to the k th class, this can be simulated by repeated drawing from

$N(\mathbf{K}_i^T \boldsymbol{\beta}_j, \sigma^2)$, $j = 1, \dots, J-1$, and accepting only when the k th component of $\mathbf{z}_i$ is the maximum.

We can see that when the number of covariates p is much larger than the number of sample size n , like in a typical gene expression microarray experiment using RKHS and Wahba representation we are able to reduce the dimension from p to n automatically. Compared to the Bayesian multinomial models proposed by Sha et al. (2004), our model does not impose a linear model structure for modelling the random latent variable. We rather consider the underlying relationship between the latent variable and the covariates to be an unknown function $\mathbf{f}$, and then use the RKHS theory to estimate that unknown function. This indeed produces a more richer class of models than before.

4.3 Bayesian Multicategory Support Vector Machine

Lee, Lin, and Wahba (2004) extended the two class SVM with the hinge loss function to the multicategory setup. In their paper they generalized the hinge loss function for binary classification to a multivariate loss for the multicategory SVM. Just as in the previous section, consider $\mathbf{t} = (t_1, \dots, t_n)$, to be n observed response data. Where t_i takes one of the J possible categories $(1, \dots, J)$. To maintain the symmetry of class label as in two class SVM, $\mathbf{y}_i$ is coded as

$$y_{ij} = \begin{cases} 1 & \text{if } t_i = j \\ -1/(J-1) & \text{otherwise.} \end{cases} \quad (4-17)$$

The separating function here will be a J -tuple function $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_J(\mathbf{x}))$. As the sum of the components of the vector $\mathbf{y}_i$ is 0, so the J -tuple function $\mathbf{f}(\mathbf{x})$ will have a sum to zero constraint, $\sum_{j=1}^J f_j(\mathbf{x}) = 0$. Let $f_j(\mathbf{x}) = \beta_{0j} + h_j(\mathbf{x})$, and $h_j(\mathbf{x}) \in \mathcal{H}_{K_j}$, for $j = 1, \dots, J$, where $\mathcal{H}_{K_j}$ denote a reproducing kernel Hilbert space of functions and h_j denote any unknown function in that functional space. Then $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_J(\mathbf{x})) \in \prod_{j=1}^J (\{1\} + \mathcal{H}_{K_j})$, the product space of J reproducing

kernel Hilbert spaces. Hence solving the multicategory SVM is finding the appropriate $\mathbf{f}(\mathbf{x})$ by minimizing the penalized multicategory loss quantity given in Equation 4-18 with the sum to zero constraint.

$$\sum_{i=1}^n L(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ + \frac{1}{2} \lambda \sum_j^J \|h_j\|_{\mathcal{H}_{K_j}}^2. \quad (4-18)$$

where, if $t_i = j$, then $L(\mathbf{y}_i)$ is a J dimensional vector with 0 in the j th component and 1 elsewhere. The $(\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+ = [(f_1(\mathbf{x}_i) - y_{i1})_+, \dots, (f_J(\mathbf{x}_i) - y_{iJ})_+]$, $(x)_+ = \max(0, x)$, and $\cdot$ denotes the Euclidean inner product. The $L(\mathbf{y}_i) \cdot (\mathbf{f}(\mathbf{x}_i) - \mathbf{y}_i)_+$ is an extension of the hinge loss function (see Section 1.4 Chapter 1) in a multicategory setup.

Lee et al. (2004) showed that, to find $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_J(\mathbf{x})) \in \prod_{j=1}^J (\{1\} + \mathcal{H}_{K_j})$, with the sum to zero constraint, minimizing Equation 4-18 is equivalent to finding $(f_1(\mathbf{x}), \dots, f_J(\mathbf{x}))$ of the form in Equation 4-19.

$$f_j(\mathbf{x}_i) = \beta_{0j} + \sum_{k=1}^n \beta_{kj} K(\mathbf{x}_i, \mathbf{x}_k | \theta); \text{ for } j = 1, \dots, J. \quad (4-19)$$

with the constraint $\sum_{j=1}^J f_j(\mathbf{x}_i) = 0$, for $i = 1, \dots, n$.

If the kernel function K is strictly positive definite, the sum to zero constraint over the data points can be replaced by the sum to zero constraint on the intercept and coefficients as in Equation 4-20 and 4-21.

$$\sum_{j=1}^J \beta_{0j} = 0 \quad (4-20)$$

$$\sum_{j=1}^J \beta_{kj} = 0, \text{ for all } k = 1, \dots, n. \quad (4-21)$$

Viewing the loss as the negative of the log likelihood, we construct our Bayesian multicategory SVM (BMSVM). The conditional distribution of $\mathbf{y}_i$ given the latent

variable $\mathbf{z}_i$ is given by Equation 4-22.

$$p(\mathbf{y}_i|\mathbf{z}_i) \propto \exp \{-L(\mathbf{y}_i) \cdot (\mathbf{z}_i - \mathbf{y}_i)_+\} \quad (4-22)$$

where $\mathbf{z}_i$ are the random J component latent vectors and $\mathbf{y}_i$'s are conditionally independent of $\mathbf{z}_i$. Just as in previous section, the latent vector $\mathbf{z}_i$ is connected to the unknown separating functions $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_J(\mathbf{x}))$ as in Equation 4-23.

$$\mathbf{z}_i = \mathbf{f}(\mathbf{x}_i) + \boldsymbol{\epsilon}_i \quad (4-23)$$

where $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \dots, \epsilon_{iJ})^T$ are the residual random vectors, $\boldsymbol{\epsilon}_i \sim N_J(0, \sigma^2 \mathbf{I}_J)$. Assuming that $\mathbf{f}(\mathbf{x})$ belongs to a product space of J RKHS and with a strictly positive definite kernel K , from Equations 4-19 to 4-21, we can write

$$z_{ij} = \beta_{0j} + \sum_{k=1}^n \beta_{kj} K(\mathbf{x}_i, \mathbf{x}_k|\theta) + \epsilon_{ij} = \mathbf{K}_i' \boldsymbol{\beta}_j + \epsilon_{ij} \quad (4-24)$$

$$\sum_{j=1}^J \beta_{0j} = 0 \quad (4-25)$$

$$\sum_{j=1}^J \beta_{kj} = 0, \text{ for all } k = 1, \dots, n. \quad (4-26)$$

We put hierarchical priors on the unknown intercepts and the regression coefficients (Equations 4-27 to 4-30).

$$\boldsymbol{\beta}_j | D_j, \sigma^2 \sim N_{n+1}(0, \sigma^2 D_j^{-1}); D_j = \text{Diag}(\lambda_{0j}, \dots, \lambda_{nj}). \quad (4-27)$$

with the constraints $\sum_{j=1}^J \beta_{0j} = 0$, and $\sum_{j=1}^J \beta_{kj} = 0, \forall k = 1, \dots, n$.

$$\sigma^2 \sim \text{IG}(\gamma_1, \gamma_2) \quad (4-28)$$

$$\theta \sim \text{U}(a_L, a_U) \quad (4-29)$$

$$\lambda_{ij} \stackrel{iid}{\sim} \text{Gamma}(c, d), \text{ where } j = 1, \dots, J, i = 1, \dots, n. \quad (4-30)$$

We notice that unlike in the case of multinomial logit model here the β_j 's are not independent. They become dependent due to the sum to zero constraints imposed on them. The joint posterior is given by Equation 4-31.

$$\begin{aligned}
 \pi(\mathbf{z}, \sigma^2, \boldsymbol{\beta}, \boldsymbol{\lambda}, \theta | \mathbf{y}) &\propto \exp \left\{ - \sum_{i=1}^n L(\mathbf{y}_i) \cdot (\mathbf{z}_i - \mathbf{y}_i)_+ \right\} \\
 &\times \frac{\exp \left\{ - \frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^{J-1} (z_{ij} - \mathbf{K}_i^T \boldsymbol{\beta}_j)^2 \right\}}{(\sigma^2)^{nJ/2}} \\
 &\times \frac{\exp \left\{ - \frac{1}{2\sigma^2} \sum_{j=1}^J \boldsymbol{\beta}_j^T \mathbf{D}_j \boldsymbol{\beta}_j \right\}}{(\sigma^2)^{(n+1)J/2} \prod_{j=1}^{J-1} |\mathbf{D}_j^{-1}|^{1/2}} \times \mathbf{I} \left(\sum_{j=1}^J \beta_{kj} = 0, k = 0, \dots, n \right) \\
 &\times \exp \left(-\gamma_2 / \sigma^2 \right) (\sigma^2)^{-\gamma_1 - 1} \\
 &\times \prod_{i=1}^n \prod_{j=1}^J \exp \left(-d\lambda_{ij} \right) (\lambda_{ij})^{c-1}.
 \end{aligned} \tag{4-31}$$

Comparing the posteriors, Equation 4-14 and Equation 4-31 we can see the main difference is in the likelihood. In the Bayesian multinomial logit model we have the multinomial likelihood, whereas in the Bayesian multicategory SVM we have the likelihood corresponding to the multivariate hinge loss. Also in Equation 4-31, sum to zero constraint on the regression coefficients are imposed using the indicator function $\mathbf{I}(\sum_{j=1}^J \beta_{kj} = 0, k = 0, \dots, n)$. Hence the posterior in Equation 4-31 has a truncated support. The conditional distribution of σ^2 , λ_{ij} , and θ is same as in (ii), (iii), and (iv) in the earlier section. The conditional posterior of $\boldsymbol{\beta}_j$, $j = 1, \dots, J$, will follow the multivariate normal as in (i) of Section (4.2), but with the sum to zero constraint imposed by Equation 4-20 and Equation 4-21. As the underlying likelihood is different in the multicategory SVM the conditional distribution (v) is now changed to $p(\mathbf{z} | \theta, \boldsymbol{\lambda}, \boldsymbol{\beta}, \sigma^2, \mathbf{y}) \propto \exp \left\{ - \sum_{i=1}^n L(\mathbf{y}_i) \cdot (\mathbf{z}_i - \mathbf{y}_i)_+ \right\} \times \exp \left\{ - \frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^J (z_{ij} - \mathbf{K}_i^T \boldsymbol{\beta}_j)^2 \right\}$. We can generate samples from the posterior Equation 4-31 by the following steps:

Step 1. Generate λ_{ij} , σ^2 , and θ exactly in the similar way as in the Bayesian multinomial logit model in the previous section.

Step 2. Generate β_j , $j = 1, \dots, J$ from the multivariate normal distribution and satisfying the constraints (Equation 4-20 and Equation 4-21).

Step 3. The conditional distribution of the latent vector $\mathbf{z}_i$ is not of standard form and we update it by blocks of $\mathbf{z}_i$ as suggested by Roberts and Sahu (1997). When $\mathbf{z}_i$ is the present state, draw a candidate value $\mathbf{z}_i^*$ from $N(\mathbf{K}_i^0 \beta, \sigma^2 I_J)$, where $\mathbf{K}_i^0 = I_q \otimes \mathbf{K}_i^T$, $\beta = (\beta_1^T, \dots, \beta_q^T)^T$. Accept the update with acceptance probability $\alpha_i = \min \left\{ 1, \frac{\exp\{-\sum_{j=1}^p L(\mathbf{y}_j) \cdot (\mathbf{z}_i^* - \mathbf{y}_j)_+\}}{\exp\{-\sum_{j=1}^p L(\mathbf{y}_j) \cdot (\mathbf{z}_i - \mathbf{y}_j)_+\}} \right\}$.

Like in our previous model, here also the RKHS gives us a low n -dimensional representation for a much higher p -dimensional problem, without making an additional projection on the feature space.

4.4 A Bayesian Gene Selection Scheme

Both of our models as described in the previous two sections has an in-built dimension reduction technique. Where a successful dimension reduction is made via RKHS and then instead of dealing with p covariates we deal with n different kernel functions. In a typical microarray experiment we have gene expression data on 5000–10,000 genes for less than 100 tumor samples. Many genes do not contain information that is useful for determining the differences between the samples. These genes should not be used for classification: indeed, sometimes they may even contain noise that can lead to incorrect classification. Although the RKHS formulation enables us to keep all the genes in our model yet an improved classification can be obtained if only differentially expressed genes are included in the model. A simple method as proposed by Dudoit et al. (2002) may be used to rank the full set of genes and select the top few genes or the genes with the most marginal relevance. But their method does not consider the possible interaction effect between several genes. In this section we propose a Bayesian variable selection technique (George and McCulloch, 1993) for our models in Section 4.2 and 4.3.

Let $X_{n \times p}$ be a matrix of gene expression data, where each column represent a gene and each row represent a sample. To do the gene selection introduce $\gamma = (\gamma_1, \dots, \gamma_p)^T$, a $p \times 1$ vector of indicators, such that

$$\gamma_k = \begin{cases} 0 & \text{the } i\text{th gene is not selected} \\ 1 & \text{the } i\text{th gene is selected.} \end{cases} \quad (4-32)$$

For a particular combination of genes or the choice of γ , X_γ denotes the reduced gene expression matrix with only those columns of the full gene expression matrix X corresponding to which the elements of γ is one. Hence the dimension of X_γ is $n \times p_\gamma$, where $p_\gamma = \sum_{k=1}^p \gamma_k$, or the number of nonzero components in the vector γ .

We put independent Bernoulli prior on γ_k , as follows:

$$\gamma_k \stackrel{iid}{\sim} \text{Bernoulli}(\omega), \quad i = 1, \dots, p. \quad (4-33)$$

The value of ω is chosen to be small to restrict the number of genes in the model. We can also include the prior knowledge of some of the genes which are more important than others by assigning different ω_k for different γ_k . A natural logic may be rather to fix the ω large and and let the model decide the sparsity. But this approach fails as in the case of microarray experiment we have very few data points, the model will fail to detect a small set of important genes and end up including many genes which are not relevant. This will lead to a poor classification performance.

Inclusion of γ , the indicator vector will result in computation of the kernel function K on the basis of the reduced expression matrix X_γ and is denoted by $K_\gamma(\cdot)$. The posterior (Equation 4-14) from the RKHS based Bayesian multinomial

logit model will now change to:

$$\begin{aligned}
 \pi(\mathbf{z}, \sigma^2, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\gamma} | \mathbf{y}) &\propto \prod_{i=1}^n p_{i1}^{y_{i1}} \dots p_{iJ}^{y_{iJ}} \\
 &\times \frac{\exp\{-\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^{J-1} (z_{ij} - \mathbf{K}_{i\boldsymbol{\gamma}}^T \boldsymbol{\beta}_j)^2\}}{(\sigma^2)^{n(J-1)/2}} \\
 &\times \frac{\exp\{-\frac{1}{2\sigma^2} \sum_{j=1}^{J-1} \boldsymbol{\beta}_j^T \mathbf{D}_j \boldsymbol{\beta}_j\}}{(\sigma^2)^{(n+1)(J-1)/2} \prod_{j=1}^{J-1} |\mathbf{D}_j^{-1}|^{1/2}} \\
 &\times \exp(-\gamma_2/\sigma^2) (\sigma^2)^{-\gamma_1-1} \\
 &\times \prod_{i=1}^n \prod_{j=1}^{J-1} \exp(-d\lambda_{ij}) (\lambda_{ij})^{c-1} \\
 &\times \prod_{k=1}^p \omega^{\gamma_k} (1-\omega)^{1-\gamma_k}.
 \end{aligned} \tag{4-34}$$

To generate samples from (Equation 4-34) in addition to all previous steps for $\mathbf{z}, \sigma^2, \boldsymbol{\beta}, \boldsymbol{\lambda}$, and $\boldsymbol{\theta}$, an additional step is required to sample the $\boldsymbol{\gamma}$. We use the conditional distribution

$$(\text{vi}) \ p(\boldsymbol{\gamma} | \mathbf{z}, \sigma^2, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \mathbf{y}) \propto \exp\{-\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^{J-1} (z_{ij} - \mathbf{K}_{i\boldsymbol{\gamma}}^T \boldsymbol{\beta}_j)^2\} \times \prod_{k=1}^p \omega^{\gamma_k} (1-\omega)^{1-\gamma_k}.$$

The above conditional distribution is not of any standard form. Hence we again deploy a MH algorithm where the components of the new update $\boldsymbol{\gamma}^*$, γ_i^* can be drawn from the Bernoulli(ω), independently. The new $\boldsymbol{\gamma}^*$ is accepted with the probability $\alpha = \min \left\{ 1, \frac{\exp\{-\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^{J-1} (z_{ij} - \mathbf{K}_{i\boldsymbol{\gamma}^*}^T \boldsymbol{\beta}_j)^2\}}{\exp\{-\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^{J-1} (z_{ij} - \mathbf{K}_{i\boldsymbol{\gamma}}^T \boldsymbol{\beta}_j)^2\}} \right\}$.

Similarly we can also incorporate the Bayesian model selection technique in our Bayesian multicategory support vector machine model. The use of the indicator vector $\boldsymbol{\gamma}$ will only change the kernel matrix, as now at each step we need to update the kernel matrix according to the genes selected in that step. Thus by this way we are not include all the genes in our model and hence adaptively eliminating the genes that are not important. The other part of the BMSVM model as explained in Section (4.3) remains same. The posterior (Equation 4-31) from the Bayesian multicategory

SVM model will now take the form

$$\begin{aligned}
\pi(\mathbf{z}, \sigma^2, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{\gamma} | \mathbf{y}) &\propto \exp \left\{ - \sum_{i=1}^n L(\mathbf{y}_i) \cdot (\mathbf{z}_i - \mathbf{y}_i)_+ \right\} \\
&\times \frac{\exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^J (z_{ij} - \mathbf{K}^T \boldsymbol{\gamma}_i \boldsymbol{\beta}_j)^2 \right\}}{(\sigma^2)^{nJ/2}} \\
&\times \frac{\exp \left\{ -\frac{1}{2\sigma^2} \sum_{j=1}^J \boldsymbol{\beta}_j^T \mathbf{D}_j \boldsymbol{\beta}_j \right\}}{(\sigma^2)^{(n+1)J/2} \prod_{j=1}^J |\mathbf{D}_j^{-1}|^{1/2}} \times \mathbb{I} \left(\sum_{j=1}^J \beta_{kj} = 0, k = 0, \dots, n \right) \\
&\times \exp \left(-\gamma_2 / \sigma^2 \right) (\sigma^2)^{-\gamma_1 - 1} \\
&\times \prod_{i=1}^n \prod_{j=1}^J \exp \left(-d \lambda_{ij} \right) (\lambda_{ij})^{\omega - 1} \\
&\times \prod_{k=1}^p \omega^{\gamma_k} (1 - \omega)^{1 - \gamma_k}.
\end{aligned} \tag{4-35}$$

Just as in Section (4.3), here also we will follow exactly same steps to generate samples from the joint posterior (Equation 4-35). The extra step needed to sample the $\boldsymbol{\gamma}$, the vector of indicators is incorporated by sampling from the conditional posterior

$$\text{(vi) } p(\boldsymbol{\gamma} | \mathbf{z}, \sigma^2, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \mathbf{y}) \propto \exp \left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^J (z_{ij} - \mathbf{K}^T \boldsymbol{\gamma}_i \boldsymbol{\beta}_j)^2 \right\} \times \prod_{k=1}^p \omega^{\gamma_k} (1 - \omega)^{1 - \gamma_k}.$$

From the conditional posterior distribution of $\boldsymbol{\gamma}$ it is clear that although the priors of the components γ_i , $i = 1, \dots, p$, assumed to be i.i.d Bernoulli(ω), but the posterior is not independent. Hence we can establish some dependency among the genes which was not possible to establish by Dudoit's (2002) criteria.

4.5 Classification of Future Cases and Identifying the Significant Genes

When we have J different classes of cancer tumors, and $J > 2$. For a new sample whose corresponding gene expression measurement is denoted by $\mathbf{x}_{new}$. The classification rule is induced by Equation 4-36.

$$\phi(\mathbf{x}_{new}) = \arg \max_j (t_{new} = j | \mathbf{x}_{new}, \mathbf{t}_{old}) \tag{4-36}$$

where

$$p(t_{new} = j | \mathbf{x}_{new}, \mathbf{t}_{old}) = \int_{\Theta} p(t_{new} = j | \mathbf{x}_{new}, \mathbf{t}_{old}, \Theta) \pi(\Theta | \mathbf{t}_{old}) d\Theta; \quad j = 1, \dots, J. \quad (4-37)$$

is the posterior predictive probability that the tumor belongs to the j th class and $\Theta = (\mathbf{z}, \sigma^2, \beta, \lambda, \theta, \gamma)$, the set of all parameters in the model. A Monte Carlo estimate of Equation 4-37 is given by Equation 4-38.

$$\hat{p}(t_{new} = j | \mathbf{x}_{new}, \mathbf{t}_{old}) = \sum_{t=1}^B \hat{p}(t_{new} = j | \mathbf{x}_{new}, \mathbf{t}_{old}, \Theta^t); \quad j = 1, \dots, J. \quad (4-38)$$

where Θ^t is t th MCMC sample from the posterior, and B is the total number of Monte Carlo samples used for estimation after the initial burn in. Hence for the new tissue sample we compute its posterior predictive probability of being in class j for all the classes $j = 1, \dots, J$ and finally assign the new sample to that class for which this probability is maximum.

For identifying the significant genes from the dormant ones we run the MCMC chain for a long time and after discarding sufficient burn in we calculate the relative number of times each gene appeared in the sample. This will serve as an estimate of the posterior probability that a single gene is included in the model, and can be used as a measure for identifying the differentially expressed genes from the inactive ones. Since we assume only the differentially expressed genes are responsible for the variation in types of tumors, and hence should be included in our model more frequently to maximize the posterior distribution. It is to be noted now that instead of a two step procedure of first gene selection and then classification our model can simultaneously select important genes and do the classification.

4.6 Applications in Multiclass Cancer Classification Using Microarrays

In this section we have applied our proposed models in Section 4.2 and Section 4.3 on two real life datasets. Bayesian gene selection criteria developed in Section

4.4 and also the gene ordering criteria as suggested by Dudoit et al. (2002) are used to include only the important genes. For both the examples we have fit in total three standard classification models : (i) Neural Network (NN), (ii) Random Forest (RF), (iii) Classical support vector machine (CSVM). None of the models, except the random forest is equipped to deal with such a high dimensional problem. In each of the cases we use the *BWS/BSS* criteria to order the genes and then selected the top few genes in each of our models. Next we fit our (iv) RKHS based Bayesian multinomial logit model (BMLM), and (v) Bayesian multicategory SVM (BMSVM). Like in (i)-(iii), in both of the models (iv) and (v) we make an initial gene selection according to the *BWS/BSS* criteria before running our model. Lastly we fit our (vi) Bayesian RKHS based Multinomial integrated with Bayesian gene selection technique (Section 4.4) (BMLM + BGS) and (vii) Bayesian multicategory SVM integrated with Bayesian gene selection technique (BMSVM + BGS). Our model (vi) and (vii) are different than (iv) and (v) in the sense that in our last two models we simultaneously predict the class label and the differentially expressed genes in a full Bayesian setup. Whereas in model (iv) and (v) the gene selection was made on the basis of *BWS/BSS* which is a strong frequentist idea closely similar to an *F*-statistic, and the classification is made on the basis of our Bayesian models.

Choice of priors play an important role in our analysis and we used near-diffused proper priors for our analysis in both the examples. The near diffuseness guarantees that our prior choice is as flat as possible but proper which ensures the propriety of the posterior along with the objectivity of our analysis. Detailed discussion on near-diffuse priors are already provided in Chapter 2 and Chapter 3. For both the examples studied, we followed the following combination of hyperparameters, $\gamma_1 = 1, \gamma_2 = 10, c = 10^{-8}, d = 10^{-5}, a_L = 0, a_U = 100$. This choice of hyperparameter is also suggested by Mallick et al. (2005), and produces a near-diffuse but proper prior. For the choice of kernel function we used the Gaussian kernel as empirically

it produces better classification result than polynomial kernel. We chose $\omega = 0.01$ so that on average only top 10% of the genes are to be included in the models producing sparsity. For all our models (model (iv) to (vii)) we have tried both multiple smoothing parameters and single smoothing parameters. The results obtained with single smoothing parameters are denoted by “*”.

We run a MCMC chain for 200,000 times and discard the first half as the burn in. To ensure that we are not stuck in one of the many modes of the posterior distribution we use multiple chains with different starting value. In both the examples studied here we used total 5 independent chains with widely different starting points and our final prediction is based on the pooled samples from these 5 chains. In both the examples we have used neural network models with 20 hidden nodes. After 20 hidden nodes we do not gain anything in terms of prediction accuracy compared to the cost of computational complexity. The *nnet* function in R with *softmax* option is used to fit the neural network model. For the random forest we have used the *randomForest* function in R with 10000 boosted trees and all other default parameters.

4.6.1 Golub Leukemia Data

The leukemia dataset is described in Golub et al. (1999). Bone marrow or peripheral blood samples are taken from 72 patients with either myeloid leukemia (AML) or acute lymphoblastic leukemia (ALL). The ALL can be further subdivided into B-cell ALL (B-ALL) and T-cell ALL (T-ALL). Thus here we have a total of 3 classes, i.e $J = 3$, AML, B-ALL, and T-ALL. Previously Mallick et al. (2005) used a Bayesian SVM model for classification of leukemia under two class setup. They considered only AML and ALL, and ignored the subclasses of ALL. Following the experimental setup of the original paper, the data are split into training and test sets. The former consists of 38 samples, of which 19 are B-ALL, 8 are T-ALL and 11 are AML; the latter consists of 34 samples, 19 are B-ALL, 1 T-ALL, and 14 are AML. The

Table 4-1: Missclassification error in the Test Set of the Leukemia data.

Methods	Genes				
	20	50	100	500	3,571
NN	2	1	3	4	-
CSVM	1	1	0	3	11
RF	1	2	2	2	3
BMLM	1	1	1	2	3
BMSVM	1	1	0	2	5
BMLM*	3	2	3	4	6
BMSVM*	2	3	4	4	6
BMLM + BGS	0				
BMSVM + BGS	0				
BMLM* + BGS	2				
BMSVM* + BGS	2				

dataset contains expression levels for 7129 human genes produced by Affymetrix high-density oligonucleotide microarrays. After Preprocessing (thresholding, filtering and base 10 log transformation) as described in Dudoit et al. (2002), we have $72 \times 3,571$ gene expression matrix.

Table 4-1 report the number of missclassification in the 34 samples of the test set. Initially we used the *BWS/BSS* criteria as proposed by Dudoit et al. (2002) to order the genes. After we order the genes we select the top 20, 50, 100, 500, and 3571 (i.e all genes with out any selection) genes and use them in the models. The first row indicates the number of top genes included in the model. We see that working initially with top 20 genes standard method like neural network gives only 2 missclassifications. Whereas random forest and classical SVM performs even better giving only 1 missclassification. Our models BMLM and BMSVM also performs equally well giving also only 1 missclassification. The main problem in

the *BWS/BSS* criteria is we do not know exactly how many top genes we should use, so we keep on adding more genes in the models. As we do so, from Table 4-1 we notice the performance of neural network decays drastically. Not only that it becomes computationally near impossible to fit a neural network model with all 3571 genes. The performance of CSVM is comparatively much better and it is computationally possible to use all genes but with 11 misclassifications. The RF works equally well for all sets of genes, validating the fact that they indeed can handle a huge volume of data. Turning our attention to our models BMLM and BMSVM we see they are very competitive with the previous standard models. For using top 500 genes they gave only 0 to 2 misclassification. Specially MBSVM correctly classifies all samples when we use top 100 genes. When all genes are included in the model their performance decays but they are much better than NN and CSVM. When we use our BMLM+BGS and BMSVM+BGS, models where we do not make any gene selection but important genes are selected adaptively using the mixture prior produces no misclassification error. Working with single smoothing parameters we observe from Table 4-1 that their performance is definitely poorer than the multiple smoothing parameter models. In all the models on average 40 genes are included in the model. Figure 4-3 plots the marginal relevance of each genes by the *BWS/BSS* criteria and also the relative number of times each gene appeared in our models. From the figure we can see that there is a significant overlap of active genes as suggested by the *BWS/BSS* criteria and our Bayesian variable selection approach. In fact there are 10 common genes as suggested by the Dudoit's criteria to be important also considered as important by our models.

4.6.2 Glioma Cancer Data

There are two main types of brain tumors: those that start in the brain (primary) and those that spread from cancer somewhere else in the body (metastasis). Primary brain tumors happen less often, and when they do, they are mostly malignant or

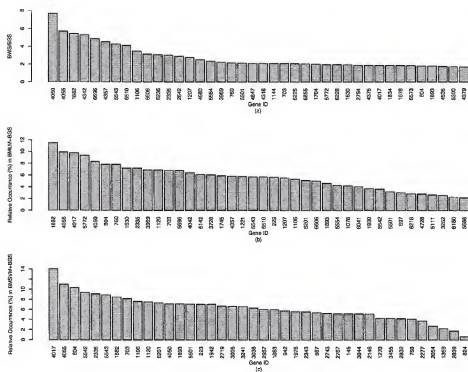


Figure 4-3: Selected genes in Golub leukemia DNA data.

(a) Marginal relevance of Genes by *BWS/BSS* criteria. (b) Relative number of times a gene is selected by our BMLM+BGS model. (c) Relative number of times a gene is selected by our BMSVM+BGS model.

cancerous. A malignant tumor is a mass or clump of cancer cells that keeps growing; it doesn't do anything except feed off the body so it can grow.

The largest group of primary brain tumors is gliomas. According to the American Brain Tumor Association, primary brain tumors occur at a rate of 12.8 per 100,000 people. Although people of any age can develop a brain tumor, the problem seems to be the most common in children ages 3 to 12 and in adults ages 40 to 70. In the United States, approximately 2,200 children younger than age 20 are diagnosed annually with brain tumors. In the past, physicians did not think about brain tumors in elderly people. Due to increased awareness and better brain scanning techniques, people 85 years old and older are now being diagnosed and treated.

There are several types of gliomas: (a) oligodendroglioma (OL), it develops from cells called oligodendrocytes that produce the fatty covering of nerve cells. This type of tumor is normally found in the cerebrum, particularly in the frontal or temporal lobes; (b) anaplastic oligodendroglioma (AO), it is a kind of faster growing oligodendroglioma; (c) anaplastic astrocytoma (AA), is the most common type of glioma and develops from a type of star-shaped cell called an astrocyte. It can occur in most parts of the brain and occasionally in the spinal cord; and (d) glioblastoma multiforme (GM), usually develops in the cerebral hemispheres, more often in the frontal lobes than the temporal lobes or basal ganglia but almost never in the cerebellum. The classification of malignant gliomas remains controversial and effective therapies have been elusive and these primary brain tumors have come under intense scientific scrutiny in recent years. Hence patient prognosis and therapeutic decisions rely on accurate pathological grading or classification of tumor cells.

The data set used here arises from glioma tissue acquired from the Brain Tumor Center tissue bank of the University of Texas M.D. Anderson Cancer Center (Kim et al., 2002). In our data we have cDNA microarray containing fragments representing 597 human genes with known functions. The tumors were diagnosed according to the

Table 4-2: Total number of missclassifications in the leave one out crossvalidation in the glioma data.

Methods	Genes			
	20	50	100	597
NN	5	4	7	-
CSVM	1	5	5	14
RF	5	6	8	10
BMLM	1	1	3	5
BMSVM	1	1	2	4
BMLM*	2	3	3	5
BMSVM*	2	2	4	6
BMLM + BGS	0			
BMSVM + BGS	1			
BMLM* + BGS	1			
BMSVM* + BGS	2			

commonly used St. Anne-Mayo criteria. The St Anne-Mayo system from 1988 is also known by the names of its authors, Daumas and Duport. The gliomas are termed according to the St. Anne-Mayo nomenclature as low grade OL, AO, AA, and GM. Hence here $J = 4$. Tissue samples are collected from 25 patients. As the number of patients is very small we use leave one out cross-validation technique to obtain the misclassification error.

Table 4-2 reports the total number of missclassifications in the leave one out cross-validation procedures. As in the leukemia data, here also we started with top 20 genes selected following the *BWS/BSS* ranking and went on adding to it and used all discussed models to predict the class of the “out of bag” sample. As we have already stated that the classification of Glioma cancer is the hardest, we observe that most of the standard methods like neural network and random forest do very poorly

in classification. Both neural network and random forest gives 5 missclassifications and as we went on adding genes to it, it produced much worse results - 7 to 8 missclassifications. The CSVM worked very well with the top 20 genes, where it misclassified in only one instance. But by increasing the number to 50 genes it gave 5 missclassifications and with addition thereafter the performance of the CSVM model diminishes drastically. Our BMLM and BMSVM models give much better result than all previous standard models. With the top 20 genes both of them gave 1 missclassification each. Also we see that as we increase the number of genes in our model to 100 their performance is not as worse as the RF, CSVM, and NN. But inclusion of all 597 genes produces 5 and 4 missclassifications. Although it is much lower than 14 missclassifications by CSVM and 10 missclassifications by RF we can definitely improve on them. Which is evident from the results of our model BMLM+BGS and BMSVM+BGS where we adaptively select the important genes and make the class predictions. For the BMLM+BGS model we commit a missclassification in a single case only, whereas BMSVM+BGS we correctly classify in all the cases of cross-validation. In both the models on average 17 genes are included. Just like in the leukemia data set, here also comparing the top suggested genes by Dudoit's criteria and our variable selection models, we see that over all 8 different genes are marked as important by both of our methods and theirs, Figure 4-4. Again this example also we see that we had a significant advantage of using multiple smoothing parameters over the single smoothing parameters in terms of missclassification error.

4.7 Discussion

In this chapter we have proposed two RKHS based nonlinear Bayesian models for classification of cancer type when we have more than two classes. The use of RKHS theory helps us to change the dimension of the problem from p to n . In cases when we use the gene expression covariates, p the number of covariates is much greater

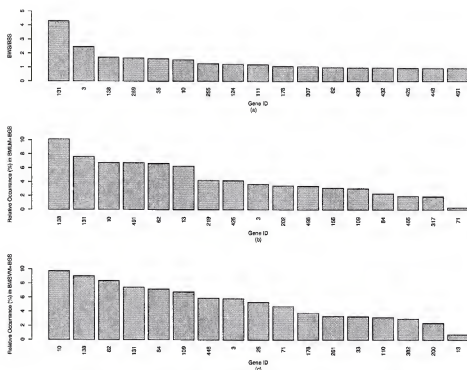


Figure 4-4: Selected genes in glioma DNA data.

(a) Marginal relevance of Genes by *BWS/BSS* criteria. (b) Relative number of times a gene is selected by our BLM+BGS model. (c) Relative number of times a gene is selected by our BMSVM+BGS model.

than n the number of samples. Hence RKHS method provides us with an automatic dimension reduction from p to n . Earlier Sha et al. (2004) proposed a Bayesian probit model approach with latent variables for modelling cancer tumors with more than two classes. But their method is much restricted compared to ours in the sense that they used simple linear model to model the latent variables. Whereas our model does not impose any kind of structure on the latent variables. The relationship between the latent variables and the microarray covariates is denoted by an unknown function f and we assumed that the function belongs to a abstract functional space, the RKHS. This is the only assumption we make. Then maximize the penalized log likelihood or the posterior to estimate the unknown function. This type of nonparametric function estimation helps us to come up with a much more flexible class of models. Although the use of RKHS helps us to bypass the problem of gene selection by already reducing the dimension of the model from p to n , but in real life applications a initial gene selection is always recommended. From Tables 4-1 and 4-2 it is clearly seen that all the standard models and our two models gave extremely poor performance when all genes are used. Rather than doing a two step model fitting, i.e, a initial gene selection and then fitting the models using only the selected genes, in Section 4.4 we suggested a integrated Bayesian gene selection and model fitting approach with the help of indicator variables.

The multicategory SVM proposed by Lin (2004) makes use of the RKHS theory and an extension of the hinge loss. Our BMSVM model is an extension of their approach in the Bayesian paradigm. Lin et al. (2004) treated the whole problem of multiclass classification in an optimization standpoint, whereas our method treats the whole problem in a probabilistic framework. Unlike the frequentist approach, in our method the kernel parameter θ is not fixed and we put a prior on it. By putting the prior on the kernel parameter we gain as we eventually use a mixture of kernels. We can also obtain the full posterior predictive probability distribution of $p(t_{new} = j)$, for

$j = 1, \dots, J$, i.e., the probability that a new tumor belongs to the j th type. The full posterior predictive probability distribution contains much more information than just a point estimate, and we can easily construct a confidence interval on it.

The use of near-diffuse proper priors helps us to make our method less sensitive to the choice of prior parameters. It also ensures that the posterior is proper so we can use all standard MCMC techniques to generate samples from it. But definitely it is sensitive to the choice of ω , as it controls the number of genes each time it is included in the model. In both the examples we have kept $\omega = 0.01$, as suggested by the works of Lee et al. (2003), and Sha et al. (2004). It means that only 10% of the genes are going to be included in the model which indirectly implies that we are inducing sparsity. We can also put a hierarchical $Beta(a, b)$ prior on ω . Interestingly from Figure 4-3 and Figure 4-4, we see that there is a significant overlap between the genes selected by our method and the genes ranked by the Dudoit's criteria of BWS/BSS . But as none of our method can be considered a full proof, so it will be more interesting to look at the genes which are labelled as important by our selection criteria but missed by them and vice versa. An insight of a biologist will be invaluable for this purpose.

In both the cases, Golub leukemia data and the glioma data our BMLM and BMSVM have lower missclassification error than the standard methods like neural network, classical support vector machine, and random forest. Only in a single instance in the leukemia data with top 100 genes the CSVM beats our BMLM model. Again here also its performance is matched by our BMSVM model. In the glioma data, our methods gave far better results than all three standard ones. Both of our methods, BMLM and BMSVM, when integrated with the Bayesian variable selection technique gives an improved performance. Hence at the end we recommend either of our models with multiple smoothing parameters augmented with the Bayesian variable selection technique.

CHAPTER 5

SUMMARY, DISCUSSION AND FUTURE RESEARCH

“You can never solve a problem on the level on which it was created.”
— Albert Einstein

In Chapters 2, 3, and 4 we have extended several distinct, but related, topics like neural networks, support vector machine, and relevance vector machine in the Bayesian paradigm. All these techniques have witnessed a tremendous growth in the last ten years. The goal of our proposed Bayesian extensions is to improve the prediction accuracy by incorporating information, a researcher knows about a problem before collecting the data into the analysis as well as to give a probabilistic view rather than just an algorithmic approach. All the models we have developed so far have a wide range of applications in machine learning, data mining and also in bioinformatics.

Neural Networks has been successfully used by the computer scientists for classification and non-linear regressions. In Chapter 2 we have extended the classical neural networks in the Bayesian paradigm. Here we put hierarchical priors on the unknown parameters of the model and end up with an entire posterior predictive distribution rather than just the point estimates of the model parameters. This helps us measure the uncertainty associated with our proposed neural network model and also improves the accuracy of prediction. The hierarchical Bayesian neural network model used for modeling bivariate binary data, has been successfully applied in prostate cancer data for staging of prostate cancer. Our proposed model jointly computes the posterior prediction probabilities of certain features indicative of non-organ confined prostate cancer like margin positivity (MP) and seminal vesicle positivity (SV). Hence it can be used by the doctors as a decision making tool in marginal cases whether to go for further diagnostic tests rather than an immediate

surgical intervention. Compared to all other methods our proposed technique has some clear advantage. Firstly, none of the previous methods could take into account the association between the two components of a bivariate binary variable. Our model with the help of a correlated error structure on the latent variables can successfully establish a dependency among the components. Secondly, we can obtain an entire posterior predictive distribution of the joint probability that a patient will develop MP and SV. Thirdly, our method can also include gene expression microarray covariates for a better prediction. In Section 2.4.2 of Chapter 2 we have shown how by going down to the molecular level we can make better staging of prostate cancer than using the standard clinical covariates. Finally, we have shown that the use of near-diffuse priors makes our hierarchical Bayesian neural network model not very sensitive to the prior choice which in turn indicates a wider applicability of our model beyond just the 3 datasets provided.

Statistical modelling and inference problems with sample sizes substantially smaller than the number of available covariates are challenging. In Chapter 3, we have introduced a full Bayesian support vector regression model with Vapnik's ϵ -insensitive loss function, based on reproducing kernel Hilbert spaces (RKHS). This provides a full probabilistic description of support vector machine (SVM) rather than an algorithm for fitting purposes. We have also considered a hierarchical relevance vector machine (RVM) where instead of using the type II maximum likelihood estimates of the hyper-parameters, we assign priors on the hyper-parameters and use Markov chain Monte Carlo technique for computation. Our proposed Bayesian SVM and RVM do not require an additional projection into sample space. The dimension reduction is automatically built into the methodology. Our models are applied for prediction of blood glucose concentration in diabetics using florescence based optics. We have also extended the full Bayesian support vector regression and relevance vector regression models when the response is multivariate. The dependency among the components of

the response vector is introduced using correlated random errors in the latent variable. The multivariate version of the SVM and RVM is applied on a prediction problem in near-infrared (NIR) spectroscopy for predicting the composition of food materials. Our Bayesian RVM and SVM has a definite advantage over all other standard methods used for modelling high dimensional data. Our method does not require any initial feature selection. For prediction purposes, our method gives much better prediction results than the current ones like partial least squares, principal component regression, stepwise multivariate linear regression, and the classical support vector machine. Our Bayesian SVM is also quite successful in handling outliers in the dataset and produces a much more stable result than any other methods. Overall, all the models developed in Chapter 3 can be considered as a new way of tackling the nonlinear regression problem with lots of covariates.

In Chapter 4, we have suggested a Bayesian alternative way of classification when the number of classes is more than two, and the covariates are far greater than the number of samples. We have proposed two models in this context, RKHS based Bayesian multinomial logit model (BMLM), and Bayesian multicategory SVM model (BMSVM). The BMSVM model is a direct extension of the multicategory SVM model proposed by Lee et al. (2004), in the Bayesian paradigm. The BMLM model can also be viewed as a classification problem with the Kullback-Leibler loss function. An immediate application of our models is made on the classification of cancer tumors using gene expression microarrays. In this chapter we have also proposed a Bayesian variable selection technique to adaptively select the differentially expressed genes from the dormant ones. To our knowledge this constitutes the first attempt to simultaneously select differentially expressed genes and tumor classification with these kinds of non linear models.

5.1 Future Research

5.1.1 Neural Networks with Variable Number of Hidden Nodes

In the neural network model suggested in Chapter 2, we have kept the number of hidden nodes M to be fixed. To find the best model we increased the nodes until we see that there is no more improvement in terms of prediction accuracy compared to computational complexity. So instead of this stepwise procedure of finding the best M , we can put a truncated Poisson or negative binomial distribution on the number of hidden nodes. And then we can build a reversible jump MCMC algorithm to do all the computations. The benefit will be that we won't need to fix the number of hidden nodes. It will be automatically selected by the model.

5.1.2 Bayesian Gene Selection Model for Neural Networks

Though the neural network is devised to solve the classification problem yet much of its success lies in, how well we can select the “important genes” that is to be included in the model. As with too many covariates or inputs it becomes very slow and computationally impossible to train a neural network. Also due to overfitting, its performance as a classifier or predictor decays in the future samples. For a typical tumor classification with microarrays, several methods like PCA and individual gene selection are used to reduce the dimensionality of the problem and enhance the performance of neural network. We are thinking of devising a stochastic search variable selection technique which can be implemented with neural network models for this purpose.

5.1.3 Nonparametric and Semiparametric Bayes Models for Tumor Classification

For all our proposed models we have used subjective priors, which leads to a lot of criticism about the prior choice. As any Bayesian model can be made to work perfectly well for a particular dataset by carefully tuning the priors. This may cause a misleading result as our analysis will be highly subjective to the prior choice and will not work well for a different dataset. Although, we used near-diffuse priors to

make our models objective but some of the model parameters still makes the analysis a bit sensitive to the prior choice. Assigning nonparametric Gaussian process priors may be one of the ways to tackle this problem, and is going to be our next line of investigation.

5.1.4 Applications in Proteomics and Spatio-temporal Data

Proteomics is the study of the proteins and their functions. SELDI-TOF (Surface Enhanced Laser Desorption/Ionization Time of Flight) and MALDI (Matrix-assisted laser desorption and ionization) mass spectrometry is a high-throughput technology to analyze protein content of a sample (e.g.: human plasma, urine). It separates the proteins according to their mass. The challenging question is: Is it possible to find patterns among the protein mass spectra of the samples that characterize and distinguish healthy individuals from diseased ones? An affirmative answer could lead to a potential diagnosis tool; additionally, the identification of proteins with different expression levels in diseased and healthy samples provide valuable insight into the pathways that underlie the disease in question. The next application of our developed models is going to be in this context.

The growing awareness of environmental issues and global change management has emphasized the need for rich, timely and flexible information systems capable of supporting decision making based on analysis of and reasoning on spatio-temporal data. Huge amounts of ecological, satellite data are now available for study. We are considering extending our models for the analysis of spatio-temporal data.

Concluding on a philosophical note...

“...But the cleverest algorithms are no substitute for human intelligence and knowledge of the data in the problem.” —Leo Breiman

REFERENCES

- Agresti, A. (2002). *Categorical Data Analysis*, Wiley, New York.
- Albert, J. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data, *Journal of the American Statistical Association* **88**: 669–679.
- Allwein, E., Schapire, R., and Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers, *Journal of Machine Learning Research* **1**: 113–141.
- Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D. and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, *Proceedings of National Academy of Science* **96**: 6745–6750.
- Aronszajn, N. (1950). Theory of reproducing kernels, *Transactions of the American Mathematical Society* **68**: 337–404.
- Bellman, R. E. (1961). *Adaptive Control Processes*, Princeton University Press, Princeton.
- Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*, Wiley, New York.
- Bishop, C. (1995). *Neural Networks for Pattern Recognition*, Claredon Press, Oxford.
- Bishop, C. and Tipping, M. (2000). Variational relevance vector machines, in C. Bouiltiler and M. Goldszmidt (eds), *Proceedings of the 16th Conference in Uncertainty and Artificial Intelligence*, Morgan Kauffman, Stanford, CA, pp. 46–53.
- Breidensteiner, E. and Bennett, K. (1999). Multicategory classification by support vector machines, *Computational Optimizations and Applications* **12**: 53–79.
- Breiman, L. (2001). Random forests, *Machine Learning* **45**(1): 5–32.
- Brown, P. J., Fearn, T. and Vannucci, M. (1999). The choice of variables in multivariate regression: A bayesian non-conjugate decision theory approach, *Biometrika* **86**: 635–648.
- Brown, P. J., Fearn, T. and Vannucci, M. (2001). Bayesian wavelet regression on curves with application to a spectroscopic calibration problem, *Journal of the American Statistical Association* **96**: 398–408.

- Buntine, W. L. and Weigand, A. S. (1991). Bayesian back-propagation, *Complex Systems* 5: 603–643.
- Chib, S. and Greenberg, E. (1995). Understanding the metropolis-hastings algorithm, *The American Statistician* 49: 327–335.
- Crammer, K. and Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines, *Journal of Machine Learning Research* 2(2): 265–292.
- Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge.
- Cybenko, G. (1989). Approximation by superposition of sigmoidal functions, *Mathematics of Control Systems and Signals* 2: 303–314.
- Denison, D., Holmes, C., Mallick, B. and Smith, A. F. M. (2002). *Bayesian Methods for Nonlinear Classification and Regression*, Wiley, New York.
- DeRisi, J. L., Iyer, V. R. and Brown, P. O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale, *Science* 278: 680–685.
- Dhanasekaran, S. M., Barrette, T. R., Ghosh, D., Shah, R., Varambally, S., Kurachi, K., Pienta, K. J., Rubin, M. A. and Chinnaiyan, A. M. (2001). Delineation of prognostic biomarkers in prostate cancer, *Nature* 412: 822–826.
- Dietterich, T. G. and Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes, *Journal of Artificial Intelligence Research* 2: 263–286.
- Dudoit, S., Fridlyand, J. and Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data, *Journal of the American Statistical Association* 97: 77–87.
- Figueiredo, M. (2002). Adaptive sparseness using jeffreys prior, in S. B. T. Dietterich and Z. Ghahramani (eds), *Neural information processing systems-NIPS*, Vol. 14, MIT Press, pp. 697–704.
- Gelfand, A. (1990). *Markov Chain Monte Carlo in practice*, Chapman & Hall, chapter Model Determination Using Sampling-Based Method, pp. 145–158.
- Gelfand, A. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities, *Journal of the American Statistical Association* 85: 398–409.
- Gelman, A. (1990). *Markov Chain Monte Carlo in practice*, Chapman & Hall, London.
- Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences (with discussion), *Statistical Science* 7: 457–511.

- George, E. and McCulloch, R. (1993). Variable selection via gibbs sampling, *Journal of the American Statistical Association* **88**: 881–889.
- Ghosh, M., Maiti, T., Kim, D., Chakraborty, S. and Tewari, A. (2004). Hierarchical bayesian neural networks: An application to prostate cancer study, *Journal of the American Statistical Association* **99**: 601–608.
- Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for gibbs sampling, *Applied Statistics*.
- Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C. and Lander, E. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science* **286**: 531–537.
- Good, I. J. (1965). *The Estimation of Probabilities. An Essay on Modern Bayesian Methods*, MIT Press, Cambridge, MA.
- Hanahan, D. and Weinberg, R. (2000). The hallmarks of cancer, *Cell* **100**: 57–70.
- Hastie, T. J., Tibshirani, R. J. and Friedman (2001). *The Elements of Statistical Learning*, Springer-Verlag, New York.
- Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Kallioniemi, O. P., Wilfond, B., Borg, A. and Trent, J. (2001). Gene expression profiles in hereditary breast cancer, *New England Journal of Medicine* **344**: 539–548.
- Huber, P. (1964). Robust estimation of a location parameter, *Annals of Mathematical Statistics* **53**: 73–101.
- Ibrahim, J. G. and Chen, M. H. (2000). Power prior distributions for regression models, *Statistical Science* **15**: 46–60.
- Ibrahim, J. G., Ryan, L. M. and Chen, M.-H. (1998). Using historical controls to adjust for covariates in trend tests for binary data, *Journal of the American Statistical Association* **93**: 1282–1293.
- Jong, S. (1993). Simpls: An alternative approach to partial least squares regression, *Chemometrics and Intelligent Laboratory Systems* **18**: 251–263.
- Kim, S., Dougherty, E., Shmulevich, I., Hess, K., Hamilton, S., Trent, J., Fuller, G. and W., Z. (2002). Identification of combination gene sets for glioma classification, *Molecular Cancer Therapy* **1**: 1153–1159.
- Kimeldorf, G. and Wahba, G. (1971). Some results on tchebycheffian spline functions, *Journal of Mathematical Analysis and Applications* **33**: 82–95.

- Law, M. H. and Kwok, J. T. (2001). Bayesian support vector regression, *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, Key West, Florida, pp. 239–244.
- Lee, K., S., N., Dougherty, E., Vannucci, M. and Mallick, B. (2003). Gene selection: a bayesian variable selection approach, *Bioinformatics* **19**(1): 90–97.
- Lee, Y., Lin, Y. and Wahba, G. (2004). Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data, *Journal of the American Statistical Association* **99**: 67–81.
- Mackay, D. J. C. (1992). A practical bayesian framework for backprop networks, *Neural Computation* **4**: 448–472.
- Mallick, B. K., Ghosh, D. and Ghosh, M. (2005). Bayesian classification of tumors using gene expression data, *Journal of the Royal Statistical Society, B* **67**: 219–232.
- McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*, Chapman & Hall, London.
- McCulloch, C. E. (1997). Maximum likelihood algorithms for generalized linear mixed models, *Journal of the American Statistical Association* **92**: 162–170.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). Equations of state calculations by fast computing machines, *Journal of Chemical Physics* **21**: 1087–1092.
- Müller, P. and Rios Insua, D. (1998). Issues in bayesian analysis of neural network models, *Neural Computation* **10**: 571–592.
- Neal, R. M. (1996). *Bayesian Learning for Neural Networks*, Springer-Verlag, New York.
- Osborne, B. G., Fearn, T., Miller, A. R. and Douglas, S. (1984). Application of near-infrared reflectance spectroscopy to compositional analysis of biscuits and biscuit doughs, *Journal of the Science of Food and Agriculture* **35**: 99–105.
- Palmgren, J. (1989). Regression models for bivariate binary responses, *Technical Report 101*, University of Washington, Seattle.
- Parzen, E. (1970). Statistical inferences on time series by RKHS methods, *Proceedings of the 12th Biennial Seminar*, Canadian Mathematical Congress, Montreal, pp. 1–37.
- Rios Insua, D. and Müller, P. (1998). Feedforward neural networks for nonparametric regression, in D. Dey, P. Müller and D. Sinha (eds), *Practical Nonparametric and Semiparametric Bayesian Statistics*, Springer-Verlag, pp. 181–191.

- Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge.
- Roberts, G. O. and Sahu, S. K. (1997). Updating schemes, correlation structures, blocking and parameterization for the gibbs sampler, *Journal of the Royal Statistical Society, B* **59**: 291–317.
- Schena, M., Shalon, D. and Davis, R. and Brown, P. (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray, *Science* **270**: 467–470.
- Schölkopf, B. and Smola, A. (2002). *Learning With Kernels*, MIT Press, Cambridge, MA.
- Sha, N., Vannucci, M., Tadesse, M., Brown, P., Dragoni, I., Davies, N., Roberts, T., Contestabile, A., Salmon, N., Buckley, C. and Falciani, F. (2004). Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage, *Biometrics* **60**(3): 812–819.
- Sollich, P. (1999). Probabilistic interpretation and bayesian methods for support vector machines, *ICANN99 - Ninth International Conference on Artificial Neural Networks*, The Institution of Electrical Engineers, pp. 91–96.
- Sollich, P. (2001). Bayesian methods for support vector machines: Evidence and predictive class probabilities, *Machine Learning* **46**: 21–52.
- Spiegelman, C., Wikander, J., O’Neal, P. and Coté, G. L. (2002). A simple method for linearizing nonlinear spectra for calibration, *Chemometrics and Intelligent Laboratory Systems* **60**: 197–209.
- Tipping, M. (2000). The relevance vector machine, in T. L. Solla and K. Muller (eds), *Neural information processing systems – NIPS*, Vol. 12, MIT Press, pp. 652–658.
- Tipping, M. (2001). Sparse bayesian learning and the relevance vector machine, *Journal of Machine Learning Research* **1**: 211–244.
- Vapnik, V. (1995). *The Nature of Statistical Learning Theory*, Springer, New York.
- Venables, W. D. and Ripley, B. B. (1997). *Modern Applied Statistics with S-Plus*, Springer-Verlag.
- Wahba, G. (1990). *Spline Models for Observational Data*, SIAM, Philadelphia.
- Wahba, G. (1999). Support vector machines, reproducing kernel hilbert spaces and the randomized gacv, in C. B. B. Schölkopf and A. Smola (eds), *Advances in Kernel Methods*, MIT Press, pp. 69–88.

- Wahba, G., Lin, Y., Lee, Y. and Zhang, H. (2002). Optimal properties and adaptive tuning of standard and nonstandard support vector machines, in D. Denison, M. Hansen, C. Holmes, B. Mallick and B. Yu (eds), *Nonlinear Estimation and Classification*, Springer, pp. 125–143.
- West, M. (2003). Bayesian factor regression models in the “large p , small n ” paradigm, *Technical report*, Duke University.
- Wold, H. (1966). Estimation of principal components and related models by iterative least squares, in P. Krishnaiah (ed.), *Multivariate Analysis*, Academic Press, pp. 391–420.

BIOGRAPHICAL SKETCH

Sounak Chakraborty was born on May 11, 1977, in Calcutta, West Bengal, India. He is the only child of Mr. Swapan Chakraborty and Mrs. Banchhita Chakraborty. After finishing his secondary and higher secondary studies from St. Lawrence High School, Calcutta, in 1996, he attended St. Xavier's College, Calcutta, under Calcutta University. From here he graduated with a Bachelor of Science degree with Honors in Statistics. He scored the highest marks in his batch of students at St. Xavier's College and was ranked third in Calcutta University. In 1999, he joined the prestigious Indian Statistical Institute (ISI), India. He received his Master of Statistics degree from ISI with specialization in Biostatistics and Data Analysis (BSDA) in 2001.

Eager to further intensify his knowledge, Sounak decided to pursue a Ph. D. degree in statistics at the University of Florida. He was awarded the prestigious Joan S. Mendenhall Fellowship in his first year of study.

In the summer of 2002, Sounak joined as a research assistant of Dr. Malay Ghosh on his project "Bayesian Neural Network Models for Bivariate Binary Data: An Application in Prostate Cancer Study." This has lead to his dissertation work in Bayesian machine learning models. Aside from completing the standard curriculum, he worked as an instructor in Fall 2004 for the course STA3032. After graduating, Sounak will be an assistant professor in the Department of Statistics at University of Missouri, Columbia.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



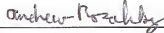
Malay Ghosh, Chair
Distinguished Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



James P. Hobert
Associate Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



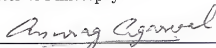
Andrew Rosalsky
Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Michael Daniels
Associate Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Anurag Agarwal
Assistant Professor of Decision and
Information Sciences

This dissertation was submitted to the Graduate Faculty of the Department of Statistics in the College of Liberal Arts and Science and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 2005

Dean, Graduate School

BAYESIAN MACHINE LEARNING

Sounak Chakraborty

(352) 392-1941

Department of Statistics

Chair: Malay Ghosh

Degree: Doctor of Philosophy

Graduation Date: August 2005

Machine learning is a computer aided way of making automatic decisions about a real life problem using some available past data. Neural Networks and Support Vector Machines are two extremely popular machine learning tools for modelling complex data. Yet they all suffer from one drawback, i.e., both of their approaches is purely algorithmic. A Bayesian way of attacking a problem involves the use of probability to represent uncertainty about the relationship being learnt. In this dissertation we have proposed a Bayesian neural network model for more efficient staging of prostate cancer. We have also proposed a Bayesian support vector machine model for prediction of composition of food materials and blood sugar concentration using near infra-red spectroscopy. Our models have shown promising improvement over all available standard ones. In this dissertation, we have also proposed two novel Bayesian models for cancer classification with the help of gene expression microarray covariates when the number of cancer types are more than two. Our proposed models are able to classify the cancer tumors more successfully in their correct group than the existing standard methods.

572 F7TH 929
10/31/05 34760